Modeling User Behavior on Socio-Technical Systems: Patterns and Anomalies

Hemank Lamba

CMU-ISR-19-106
December 2019

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jürgen Pfeffer (Co-Chair)
Christos Faloutsos (Co-Chair)
J. Zico Kolter
Ceren Budak (University of Michigan)

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

This research was sponsored by the Army Research Office under Award No. W911NF14010481, LexisNexis, HPCC Systems, National Science Foundation under grant numbers CNS-1314632, IIS-1408924, CAREER-1452425, IIS-1408287, the US Army Research Lab under grant number W911NF-09-2-0053, National Security Agency Science of Security Lablet grant, the Carnegie Mellon Presidential Fellowship, the Snap Inc. Fellowship and the Adobe University Marketing Research Award. 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: data mining, user behavior modeling, computational social science, anomaly detection, social networks, social media

To my parents and maama,
for their love and support.

Abstract

How can we model user behavior on social media platforms and social networking websites? How can we use such models to characterize behavior on social media and infer about human behavior and preferences at scale? Specifically, how can we describe users that indulge in posting about risk-taking behavior on social media or mobilize against a particular entity in a firestorm event on Twitter?

Online social network platforms (e.g. Facebook, Twitter, Snapchat, Yelp) provide means for users to express themselves, by posting content in the form of images and videos. These platforms allow users to not only interact with content (liking, commenting) but also to other users (social connections, chatting) and items (through ratings and reviews), thus providing rich data with huge potential for mining unexplored and useful patterns. The availability of such data opens up unique opportunities to understand and model nuances of how users interact with such socio-technical systems, while also contributing novel algorithms that can predict genuine user behavior and also detect malicious entities at such a large scale.

In this dissertation, we focus on two broad topics - (a) understanding user behavior on social media platforms and (b) detecting fraudulent activities on these platforms. For the first part, we concentrate on user behavior in two different settings - (i) individual user behavior, where we analyze behavior of actions taken at individual scale for example modeling how does individual's expertise in e-commerce systems (such as wine rating, movie rating) evolve over time? and how can that be used to recommend the next product? The second sub-part (ii) focusses on user-based phenomena, where multiple users are analyzed collectively to discover an interesting phenomena, for example what are the characteristics of communication pattern between users participating in a firestorm event. In the second setting, we tackle the problem of detecting fraudulent activities on social media platforms. We solve two related sub-themes in the problem area, in the first sub area, we characterize various fraudulent activities on social media platforms and propose anomaly detection models to identify fraudulent users and activities. For the next sub-area we propose models that are not only confined to social media platforms, but can also be extended to general settings. Overall, this thesis looks at two closely related problems i.e. modeling user behavior on social media platforms, and then using similarly generated models to detect abnormal and potentially fraudulent behavior.

Acknowledgments

First, I want to thank my parents for their immeasurable support. Their uncountable sacrifices for me have played a pivotal role in me trying to achieve my dreams. Navigating tough times during the past 5 years and my entire life was only possible due to their unconditional love and support. Anything that I am, or am able to achieve in my life is because of them. Thanks, mom and dad, this is for you.

This thesis would not have been possible if it wasn't for my advisors. I want to thank Jürgen Pfeffer for placing his trust in me and taking me as his 2nd Ph.D. student. His constant advice and support on both research topics and also on personal life have helped me immeasurably navigate the graduate school life. I cannot acknowledge much the number of times he went to fight for me with different scenarios. He has taught me to be an independent researcher, and more broadly a friendly and kind person. When I encountered severe challenges during my study, Christos Faloutsos emerged a savior, to whom I am eternally grateful. I want to thank him for his extreme patience and persistence with me as a young graduate student. He has played an integral role in shaping me as a researcher. I will miss working with him as a student - and for now, I can just reminisce about the late nights where he will be available till the deadline, the charades game played at every gathering, or even the snacks that always welcomed you as soon as you enter his office. I would also like to thank my other committee members, Zico Kolter and Ceren Budak. Thank you both for your comments, feedback and guidance that significantly shaped this thesis.

During my Ph.D., I was lucky to work with multiple mentors. At the time when both Christos and Jürgen were unfortunately remote, I got a chance to work with Leman Akoglu. I want to thank her for support during this time of Ph.D. life. Her hardworking style and attention to detail, combined with clear communication of difficult concepts made me strive to become a better student. Another time, when I was lost and in seek of a mentor - Bogdan Vasilescu emerged as a savior. Working with him made me rediscover and strengthen the reason why I wanted to be a researcher in the first place. He not only ensured that I go on to complete my Ph.D. but also was there for me for every obstacle I faced during the last couple of years. I want to thank Bogdan not only for helping me just on the academic front but also providing me immense support and confidence to just be able to navigate the hard times even in personal life. I also want to thank Jim Herbsleb, former director of the Societal Computing program (and now the director of ISR!) for supporting me in more than one way during my stint at CMU. I could not emphasize enough that if Jim wasn't there, it would have been very likely that I would have dropped out.

One of the most exciting parts of my time in graduate school has been my summer internships. I would like to thank Rayid Ghani for selecting me as one of the Data Science for Social Good Fellows for 2015. The summer in Chicago was probably one of the best summers I have spent, interacting with a lot of bright minds from various disciplines working on interesting data science problems. I would like to thank Kush Varshney, Aleksandra Mojsilovic, and Mary Helander for giving me a chance to spend summer at IBM Research as part of the inaugural class of Science for Social Good fellows. Their support and guidance were helpful enough for me to be able to make the summer productive even though the project did run into some unforeseeable challenges. In summer 2017, I worked at Google with Senaka Buthpitiya. I want to thank him for providing me the chance to work in his team and help me learn some valuable engineering skills and apply my research to a novel setting, a setting on which I have never worked on before, thus enabling me to expand the applicability of my work. Lastly, I want to thank my very good friend from graduate life and mentor - Neil Shah for offering me a chance to intern at Snap Research. The entire summer in Los Angeles was one of the most fun and productive ones, where I get a chance to learn a lot from his expertise both professionally and even at a personal level. When I look back at the summer, I think the best part was the long unstructured discussions on the future of data mining research on Venice beach with coffee from Menotti's.

It would be extremely unfair on my part to not thank my undergraduate mentors - the ones who introduced me to research. I published my first paper with Mayank Vatsa and Richa Singh. I thank them deeply for introducing and teaching me ropes of how to do research - they were extremely patient with me and trusted me to carry out an entire research-level project. I especially want to thank Ponnurangam Kumaraguru (PK) for being a great mentor, who has still stayed in touch and continues to collaborate. Some of my most cited, most well-known papers have come while working with PK, and it is hard to ignore his super friendly and get-it-done attitude. He has constantly motivated me to become a better version of myself. I am quite certain that he has a massive role to play in my career. I would also like to thank all the current and former students of PK's research group (PreCog), from whom I constantly keep learning. I especially want to thank Aditi Gupta, who trusted me to mentor me. After my undergraduate, I spent wonderful and intense 2 years at IBM-Research. I want to thank Amit Nanavati (dude) for being an amazing and supportive mentor and a friend to me not only through the IBM phase but also through my Ph.D. Ramasuri Narayanam was another mentor who taught me the ropes of doing and writing a research paper. I want to thank him for trusting me with some cool research problems.

I am very thankful for the colleagues and friends that I have gotten to know during my graduate studies. I especially want to thank Momin Malik. Momin was a great mentor to me and through the countless conversations we had, I grew up to become a good researcher. I am equally thankful to another great mentor - Neil Shah, with whom I got so many opportunities to work and each time it was a great learning experience. I am also thankful to the members and visitors of the CMU database group. I want to thank Alex Beutel, Vagelis Papalexakis, Neil Shah, and Bryan Hooi for welcoming me to the group and mentoring me as a young student. I also want to thank Dhivya Eswaran, Kijung Shin, Hyun Ah Song, Minji Yoon, and Srijan Kumar for fostering a friendly environment and being a lot of fun to hang out with. I am also grateful to Emaad Manzoor, in whom I found a great collaborator. Working with Bogdan, I also got to interact with his group (Strudel) - I want to thank all of them (Shurui Zhou, David Widder, Hongbo Fang, Asher Trockman, Jeremy Lacomis, Daye Nam, Marat Valiev, Sophie Qiu, Chris Bogart) for being so welcoming to me at all their group activities and make me feel at home.

During my graduate studies, I got a chance to collaborate with a lot of amazing people - Constantine Nakos, T.J. Glazier, Bradley Schmerl, David Garlan, Subhabrata Mukherjee, Gerhard Weikum, Vaishnavh Nagarajan, Naji Shajarisales, Niharika Sachdeva, Megha Arora, Varun Bharadwaj, Divyansh Agarwal, Mayank Vaccher, Vedant Nanda, Shreya Jain, Dipankar Niranjan, Shashank Srikant, Dheeraj Reddy, Karandeep Juneja, Shwetanshu Singh, Asher Trockman, Daniel Armanios, Heather Miller, Daniel Klug, Bogdan Vasilescu, Claudia Mller Birn, Katja Mayer, Neil Shah, Bryan Hooi, Kijung Shin, Emaad Manzoor, Momin Malik, Ponnurangam Kumaraguru, Juergen Pfeffer and Christos Faloutsos. I thank all of them - I learned something valuable from all of them.

Beyond those in the research groups and coauthors, there are academic friends (not collaborators, yet!) with whom I had long interactions at various conferences and I cannot thank them enough for making my conference experience all the more fun – Kenneth Joseph, Daniel Romero, Ivan Brugere, Aditya Parkash, Bijaya Adhikari, Martin Saveski and Caitlin Kuhlman.

I am grateful for many amazing colleagues at ISR who were always there to listen to me and help me out. First and foremost in this list are my officemates, who will bring me amazing sweets and candies from their home and travels - Shurui Zhou and Jaspreet Bhatia. We were quite lucky to have all 3 new students paired up in the same office – we navigated through a lot of professional and personal struggle together. I want to thank Gabriel Ferreira, Morgan Evans, Tom Maglensiki for being great fun and super cool to hang out with. I want to thank Hanan Hibshi and Hana Habib for helping me out at various stages of my graduate studies. Beyond ISR, I got to interact with amazing students without whose help I would have never been able to understand some of the hard concepts (or even pass the course) — Naji Shajarisales, Manu Reddy, Aman Gupta, Guillaume Lample, Jakob Bauer, Krishna Pilluta, William Herlands, Maria De Arteaga, Maria Cuellar, Zhe Zhang, Veronica Marotta, Wenting Yu, and Rishav Chakravarti.

It cannot be emphasized the role friends play in your lives - and I was very very fortunate to have some amazing set of friends throughout my course of Ph.D. First and foremost, I want to thank my roommates and others who just hung out so much at our place that they were like roommates – Rahul Muthoo, Kirtikar Kashyap, Aditya Shantanu, Varun Saravagi, Yasha Jain, Harsha Rastogi, Ishant Dawer. They were some of the first friends that I made in Pittsburgh, and they provided the dose of fun I required in the first 2 years of my graduate school. It is not an overstatement but it was never a dull moment hanging out with them. After they all (it was hard to believe but they all did!) graduated, there was a huge void left in my social life - but it was immediately filled by another bunch of equally fun folks that I met. Shreyash Srivastava, Ritesh Agarwal, Nandan Pitre, and Vanshaj Sikri, who have been like brothers I never had - their presence in my life made harsh winters and humid summers of Pittsburgh very bearable. They reminded me that there is life beyond CMU, and helped me maintain a good work-life balance. During my stint at Chicago, I made friends who have had a lasting effect on me as they introduced me to hobbies and activities that I had never experienced before – Fridolin Linder, Julian Katz Samuels, Benedict Kuester, and Esha Maharishi.

I am grateful to also the friends that are in my hometown, who supported me through all the ups and downs, and also ensured that I could give a shot to pursuing Ph.D. Knowing that there are some of my friends who are in Delhi, who are there for my parents in case of emergency, is all that I found strength in and pursue my education here. First and foremost, I would like to thank Puroo Soni, Vikram Sharma, and Rohit Madan with whom I grew up and who have been there with me for almost every step. Additionally, I want to thank my college mates, who always have provided me a great safety net for when I fell – Abhishek (Logan), Sumit Arora, Aman Sahni, Prateek Gaur, Saurav Maitra, Divij Wadhawan, Stuti Ajmani, Tanushree Mishra and Charvi Puri.

This journey would have not been possible without the support of some amazing staff at SCS. First, I would like to thank Connie Herold and Nick Frollini for supporting me when I was about to tap out and ensuring that I do indeed cross the finish line. I especially want to thank Connie Herold for being an absolute positive force in my journey, and going to fight for me more than once. I also want to extend my gratitude to Sharon Blazevich, who is a great and warm figure for all of the students in ISR. She is very welcoming, and her office was a constant source of all great cookies, pastries, and cakes. Most of my conference reimbursements were handled by Ann Stetser, who was very patient with me and my weird passport/visa issues and helped me breeze through all of them very effectively, for which I cannot thank her enough.

When things went south, I often looked towards friendly faces at CAPS - and I am thankful to them for helping me through various bad segments of my life. Graduate student's life can be very lonesome - I was glad to be part of Dec/5 and was able to work with some amazing fun graduate students (Sudarshan Wadhkar, Vittorio Perera, Carla De Viegas, Chirag Nagpal, Sree, Shreyash Srivastava) and obviously the backbone of Dec/5 - Catherine Copetas to organize social events for all of the graduate students at SCS.

Last but not least, I want to thank my maama and maami, who have been like the second set of parents to me. Probably one of the only person I looked up to, while growing up was my maama. There was a time when I might have dropped out in my undergraduate due to financial constraints - but he was there ready to help me. Unfortunately, he passed away during the last year of my Ph.D. I will forever live with the regret of him not being able to see what I have achieved or will achieve. This thesis is just a small token of gratitude towards him.

CONTENTS

I Introduction and Background 1

1 Introduction 3
1.1 Overview and Contributions 4
1.1.1 Characterizing User Behavior 4
1.1.2 Anomaly Detection 6
1.2 Thesis Organization 7

2 Preliminaries and Background 11
2.1 Online Social-Technical Platforms 11
2.2 Graphs 12
2.3 Tensors 13
2.4 Learning 14

II User-based Phenomena on Social Media 15

3 Measuring the impact of firestorms 17
3.1 Background and Related Work 18
3.1.1 Firestorms 18
3.1.2 Twitter 20
3.1.3 Specific firestorms 20
3.2 Methodology 21
3.2.1 Firestorm Identification 21
3.2.2 Firestorms collection 21
3.2.3 Data Source 22
3.2.4 Data Extraction 22
3.3 Results 24
3.3.1 Firestorms 24
3.3.2 Case studies 25

3.3.3 Mention networks . . . 28
3.4 Discussion . . . 30
3.5 Conclusion . . . 30

4 Understanding bias in geocoded data **31**
4.1 Background and Related Work . . . 32
4.2 Data Collection . . . 34
4.2.1 Geo-Coded Twitter Data . . . 34
4.2.2 Geospatial Data . . . 35
4.2.3 Socioeconomic Data . . . 36
4.2.4 Mobile users . . . 36
4.3 Statistical Models . . . 37
4.3.1 Random distribution over population . . . 37
4.3.2 Model specification . . . 38
4.3.3 Spatial errors model . . . 39
4.4 Results and Discussion . . . 39
4.4.1 Observational Results . . . 39
4.4.2 Bivariate regression model . . . 41
4.4.3 Spatial errors model . . . 42
4.5 Conclusion . . . 45
4.6 Future Directions . . . 46

III Individual User Modeling **49**

5 Detecting dangerous selfies behavior on social media **51**
5.1 Related Work . . . 53
5.2 Selfie Deaths Characterization . . . 54
5.3 Selfie Dataset Curation . . . 56
5.4 Feature Set Generation . . . 58
5.5 Experiment . . . 62
5.5.1 Manual Annotation . . . 62
5.5.2 Classifier . . . 63
5.6 Conclusions . . . 66

6 Detecting distracted driving posts on social media **69**
6.1 Development of Research Questions . . . 71
6.2 Data Collection and Dataset . . . 72
6.2.1 Data Collection . . . 73
6.3 Detecting Distracted Driving Content . . . 74
6.4 Characterizing Temporal and Spatial Patterns . . . 77
6.4.1 Extent of distracted driving content . . . 77
6.4.2 Temporal Analysis . . . 77
6.4.3 Spatial Analysis . . . 79

6.5 Characterizing Users . . . 81
6.5.1 Explanatory Variables . . . 81
6.5.2 Effect of Variables . . . 81
6.5.3 Statistical Model . . . 82
6.6 Discussion . . . 84
6.6.1 Research Questions . . . 84
6.6.2 Implications . . . 85
6.6.3 Threats to Validity . . . 86
6.7 Related Work . . . 87
6.8 Conclusions . . . 87

7 Modeling experience in recommendation systems 89
7.1 Overview . . . 92
7.1.1 Model Dimensions . . . 92
7.1.2 Hypotheses and Initial Studies . . . 93
7.2 Building Blocks of our Model . . . 94
7.2.1 Latent-Factor Recommendation . . . 95
7.2.2 Experience-based Latent-Factor Recommendation . . . 95
7.2.3 User-Facet Model . . . 95
7.2.4 Supervised User-Facet Model . . . 96
7.3 Joint Model: User Experience, Facet Preference, Writing Style . . . 97
7.3.1 Generative Process for a Review . . . 97
7.3.2 Supervision for Rating Prediction . . . 98
7.3.3 Inference . . . 99
7.4 Experiments . . . 102
7.4.1 Quantitative Comparison . . . 103
7.4.2 Qualitative Analysis . . . 104
7.5 Use-Case Study . . . 106
7.6 Related Work . . . 108
7.7 Conclusion . . . 109

IV Identifying Fraud on Social Media 111

8 Understanding Link Fraud services 113
8.1 Related Work . . . 114
8.2 Know Thy Enemy: Characterizing Link Fraud . . . 115
8.2.1 Setup and Data Collection . . . 116
8.2.2 Network Observations . . . 119
8.2.3 Attribute Observations . . . 125
8.3 Assessing Discriminative Power of Entropy Features . . . 128
8.4 Discussion . . . 130
8.5 Conclusion . . . 130

9 Modeling Dwell Time Engagement Fraud 133
9.1 Related Work . . . 135
9.2 Data Description . . . 136
9.3 Initial Observations . . . 138
9.4 Individual Dwell Time Modeling . . . 139
9.4.1 Multimedia Content Modeling . . . 139
9.4.2 Viewers . . . 144
9.5 Aggregate Dwell Time Modeling . . . 146
9.5.1 Copula Modeling . . . 147
9.5.2 Multimedia Content . . . 148
9.5.3 Viewers . . . 148
9.5.4 Validation . . . 149
9.6 Anomaly Detection . . . 151
9.6.1 Robustness to contamination . . . 151
9.6.2 Effectiveness on real data . . . 152
9.7 Scalability . . . 152
9.8 Conclusion . . . 153

10 Detecting Chatbots on Livestreaming applications 155
10.1 Related Work . . . 157
10.2 Problem Statement . . . 158
10.3 Data Description . . . 159
10.4 Initial Observations . . . 160
10.5 Proposed Framework: SHERLOCK . . . 162
10.5.1 Stage I: Detecting Chatbotted Streams . . . 162
10.5.2 Stage II: Detecting Constituent Chatbots . . . 163
10.6 Experiments . . . 165
10.6.1 Baselines . . . 165
10.6.2 Results on Real Dataset . . . 165
10.6.3 Synthetic Dataset Generation . . . 166
10.6.4 Results on Synthetic Dataset . . . 167
10.6.5 Scalability . . . 168
10.7 Conclusions . . . 168

V Anomaly Detection Beyond Social Media 171

11 Individual metrics in group-based temporal fraud detection 173
11.1 Related Work . . . 177
11.2 Preliminaries and Problem Definition . . . 178
11.2.1 Problem Definition . . . 178
11.2.2 Block Level Suspiciousness Metrics . . . 178
11.2.3 Axioms . . . 179
11.2.4 Shortcomings of Other Metrics . . . 180

11.3 Proposed Approach:ZOORANK . . . 180
11.3.1 Temporal Feature Handling . . . 180
11.3.2 Proposed Metric . . . 181
11.3.3 Algorithm . . . 181
11.4 Experiments . . . 183
11.4.1 Datasets . . . 183
11.4.2 Q1. Effectiveness of ZOORANK . . . 184
11.4.3 Q2. Generalizability of ZOORANK . . . 185
11.4.4 Q3. Scalability of ZOORANK . . . 188
11.5 Conclusions . . . 188

12 Incorporating human feedback for anomaly detection **191**
12.1 Related Work . . . 194
12.2 Preliminaries and Problem Definition . . . 196
12.2.1 Learning on-the-job Setup . . . 196
12.2.2 Family of Detection Models . . . 196
12.2.3 Metrics of Interest and Problem Statement . . . 197
12.3 Proposed Approach: OJRANK . . . 198
12.3.1 Generating pairs . . . 198
12.3.2 Optimization . . . 199
12.4 Evaluation . . . 201
12.4.1 Baselines . . . 201
12.4.2 Datasets . . . 203
12.4.3 Results . . . 204
12.4.4 Sensitivity Analysis . . . 208
12.5 Conclusion . . . 209

VI Conclusions and Future Work **211**

13 Conclusions **213**
13.1 Contributions . . . 213
13.1.1 User-based Phenomena on Social Media . . . 213
13.1.2 Individual User Modeling . . . 214
13.1.3 Identifying Fraud on Social Media . . . 214
13.1.4 Anomaly Detection Beyond Social Media . . . 215
13.2 Impact . . . 215
13.2.1 Academic Impact . . . 215
13.2.2 Practical Impact . . . 215

14 Future Work **217**
14.1 Computational Social Science . . . 217
14.1.1 Causal Inference . . . 217
14.1.2 Polarization and Social Media . . . 218

14.2 CyberSecurity 218
14.2.1 Adversarial Data Mining 218
14.2.2 Human-in-the-loop anomaly detection 218
14.3 Extending to other domains 219

Bibliography **221**

LIST OF FIGURES

1.1 Proposed method [158] proposes (a) state-of-the-art parametric models for individual sample dwell times for content which closely mirror empirical data, (b) flexible copula modeling of aggregated multivariate parameter fits, (c) utilization of aggregate models for detecting dwell time engagement anomalies which (d) reflect abnormal behaviors radically inconsistent with most samples. 7

2.1 Illustration of a 3-mode Tensor $\boldsymbol{\mathcal{X}}$ having dimensions $I_1 \times I_2 \times I_3$, with a subtensor $\boldsymbol{\mathcal{Y}}$. 13

3.1 Histogram of peak sizes of collected firestorms, with a scaled fitted logspline. The x-axis is the estimated volume of tweets reached on the peak day, and the y-axis is the number of firestorms reaching that volume. 22

3.2 Histogram of distribution of the number of days it took the collected firestorms to decay to 90% of peak volume, with scaled fitted density. 23

3.3 Boxplot of peak volume versus the days it took to decay to 90%. 24

3.4 Estimated number of tweets with hashtag #myNYPD (case-insensitive) over 16 days in 2014, plotted at the midpoints of bins with a width of 16.67 minutes. Ticks are at 6 hour intervals, UTC. The number of tweets from users using the hashtag for the first time is used to measure the number of new users. Since the total number of tweets is very close to this number, we include the difference between the two, which are the tweets from returning users. 25

3.5 Estimated number of tweets with hashtag #CancelColbert (case-insensitive) over 16 days in 2014, with other details identical to that in fig. 3.4. 27

3.6 Distribution of tweets per user in the decahose (log-log scale). 28

3.7 Network of mentions between firestorm participants, in this case for #askJPM, aggregated by week, before, during, and after the event. 28

3.8 Distributions of the Jaccard index of edges between the mention networks two weeks before, one week before, during, one week after, and two weeks after the firestorms. Vertical lines are put at the mode of each distribution. The matrix is symmetric; this redundancy is provided for ease in vertical comparisons. The figure shows that the networks after the firestorms resemble much more the networks before the firestorm than during the firestorm. 29

4.1 Quintiles of population per square mile by 'block group' (see below) in the 2010 Decennial Census. 33

4.2 Quintiles of geotag users, uniquely assigned (see 'mobile users' below) per block group, divided by block group area. 33

4.4 The usual long-tailed distribution of the number of users who have tweeted a certain number of tweets. Because of this skew, we focus on unique users alone, and ignore the volume of tweets. 35

4.5 A full 77.61% of geotag users in our set tweeted only from one state, and having tweeted from 5 or fewer states accounts for 99.21% of users. 41

4.6 34.76% of geotag users tweeted only from one block group. 27 or fewer block groups were 95%, 50 or fewer block groups were 99%. One outlier at 23,547 excluded. 42

4.7 Eliminating zero-count observations reduces the artifacts visible at $x = 0$ and $y = 0$ but does not substantially change the fit. 43

4.8 The relationship between males and total population behaves exactly as we expected of a quantity randomly distributed over the population, making it an effective null model against which to compare the observed distribution of geotag users. 44

5.1 Dangerous Selfies 52

5.2 A brief overview of our approach - Tweets tagged with a geolocation are analyzed using text, location and image-based features. Whereas tweets without a geolocation are analyzed only using text and image-based features. 53

5.3 (a) Number of Deaths due to various reasons, and (b) Number of Incidents. . . . 56

5.4 CDF Plots showing the difference in the distribution of height-related features for dangerous and non-dangerous images. Left: Maximum Elevation in 5km radius and 5 sampled locations (p-value:0.028). Center: Maximum difference in elevation of 10 points sampled in 1km radius with the elevation of the location (p-value: 7.09e-6). Right: Maximum Elevation Difference of 10 points sampled in 1km radius (p-value: 1.22e-9). 58

5.5 Segmentation Example: Different stages of processing to get the final segmented image distinguishing between the water and land. 60

5.6 CDF Plots showing the difference in dangerous and non-dangerous distributions for water-related features. Left: Minimum distance to a water body. Right: Fraction of water pixels in the segmented image 60

5.7 An example of the DenseCap on one of the images (Left) from our dataset. We use the dense captions produced by DenseCap (Right) to come up with text based features over them. 61
5.8 t-SNE scatter plot of doc2vec output of generated captions for 50 randomly chosen dangerous and non-dangerous selfies. 62
5.9 Screenshot of the annotation tool. We asked above questions to the annotators based on a selfie image shown to them. 64
5.10 Receiver Operating Characteristic (ROC) curves corresponding to the statistical models for identifying dangerous selfies. "Dangerous" selfie is the positive class. 66

6.1 Precision and Recall for distracted driving class for (a) Random frame and (b) Single frame for different thresholds. 76
6.2 The top 30 cities in our dataset ordered based on the ratio of driving snaps to the total snaps. 78
6.3 (a) Diurnal trends (for both the driving and non-driving content classes). The line plots denote the regression fit of the trends. (b) Cities clustered according to their temporal patterns. 79
6.4 Spatial analysis (frequency distribution plots) of three cities (from Table 6.1) . . . 79
6.5 (a) Power-Law distribution fits the best for most of the cities, in comparison to other candidate distributions. (b) Sample fits under Power-Law distribution shown for (top) Riyadh and (bottom) Delhi. 80
6.6 Scatter plot of how number of driving snaps is affected by different variables: (a) Gender Ratio: Ratio of Males to Females (b) Development status of the city (c) Population of the city, (d) Ratio of population between ages 0 and 20 82

7.1 KL Divergence as a function of experience. 94
7.2 Supervised model for user facets and ratings. 96
7.3 Supervised model for user experience, facets, and ratings. 97
7.4 MSE improvement ($\%$) of our model over baselines. 103
7.5 Proportion of reviews at each experience level of users. 106
7.6 Facet preference and language model KL divergence with experience. 107

8.1 Comparison of network and attribute features across user types 115
8.2 Egonet differences in freemium and premium fraud 120
8.3 Boomerang network differences in freemium and premium fraud 123
8.4 Differences across user types' follower action counts 127
8.5 Comparison of descriptive word-use between fraud types 128
8.6 Entropy-based features show strong classification performance 129

9.1 Our work discusses (a) state-of-the-art parametric models for individual sample dwell times which closely mirror empirical data, (b) flexible copula modeling of aggregated multivariate parameter fits, (c) utilization of aggregate models for detecting dwell time engagement anomalies which (d) reflect abnormal behaviors radically inconsistent with most samples. 134

9.2 Median dwell time ratios vs. number of views on (a) unlooped and (b) looped media, and (c) viewers show outliers which exhibit excessively high dwell times compared to normal engagement patterns of similar view-count peers. 135

9.3 Aggregated dwell time ratio statistics for varying media types and durations inform our modeling choices: treat images and videos similarly, and unlooped and looped content distinctly. 137

9.4 LM-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the $45°$ line, better the fit. (b) %age of samples where model fits were successful($p < .05$). 140

9.5 Proposed LM-DP (red) visually matches empirical dwell times (blue) across several looped media samples of varying patterns. 141

9.6 Our proposed UM-DP (red) visually matches empirical dwell time probabilities (blue) across unlooped media samples with varying viewing patterns. 143

9.7 UM-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the $45°$ line, better the fit. (b) %age of samples where model fits were successful($p < .05$). 143

9.8 Our proposed V-DP (red) visually matches empirical dwell times (blue) across several looped media samples with varying viewing patterns. 144

9.9 V-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the $45°$ line, better the fit. (b) %age of samples where model fits were successful ($p < .05$). 146

9.10 Bivariate and C-vine copula structures can model joint densities parametrically. (a) and (b) show our LM-AM and UM-AM dependency structures, respectively. . 149

9.11 **Aggregate C-vine models closely approximate real data**. Pairwise dependency heatmaps between original data (top) and simulated data (bottom) are visually close. 150

9.12 C-vine models are robust and consistent over time. Pairwise dependency heatmaps between simulated data from aggregate models trained on two different months (top and bottom) are visually close. 151

9.13 Our aggregate models detect real dwell time anomalies. The subplots show huge disparities in the mean dwell time ratio distributions between anomalous and normal (a) unlooped media, (b) looped media and (c) viewer samples. 153

9.14 Our model inference is scalable: (a-c) show that individual fitting, copula preprocessing via integral transform, and copula inference are all near-linear in sample size. 153

10.1 (**Left**) Livestreaming platforms offer chatrooms (top), which streamers can manipulate via chatbotting tools (bottom) that enable customization of chat interval, number of chatters and even message contents. (**Center**) We propose SHERLOCK, a two-stage chatbot detection approach based on stream (top) and user-level classification (bottom). (**Right**) We enable discovery of chatbotted streams (top – notice genuine users asking for moderators to handle the bots), and the constituent chatbots via discriminative features (bottom – large points indicate high user density). 156

10.2 (a): ECDF for median distribution on number of messages for genuine and chatbotted streams. (b): Distribution of number of messages posted for randomly selected genuine and chatbotted streams. 160

10.3 (a): ECDF for distribution of median on IMD for genuine and chatbotted streams. (b): Distribution of IMD. 161

10.4 (a): ECDF for distribution of median on number of windows per user for genuine and chatbotted streams. (b): Distribution of number of windows per user. 162

10.5 Performance of SHERLOCK on various attack models (bar colors), stream durations (bar groups), noise levels (columns) and noise types (bot users in (a-c), and bot messages in (d-f)). SHERLOCK is robust to noise and performs consistently well across varying adversarial configurations, with F1 scores generally over 0.80. 168

10.6 SHERLOCK has near-linear runtime in (a) # streams (Stage I) and (b) # users (Stage II). 169

11.1 **Effectiveness of ZOORANK on real world datasets.** (**Top Left**) Perfect precision-recall on software marketplace dataset. (**Top Right**) ZOORANK obtains good precision recall on Reddit dataset. (**Bottom Left**) Top 100 suspicious users found by ZOORANK show high synchronicity (formed groups) in rating and reviewing top suspicious products. (**Bottom Right**) The suspicious users (bottom; red) detected by ZOORANK for Reddit dataset show irregular spikes in inter-arrival time distribution, as compared to all the users (top; blue). 175

11.2 How to rank users based on their suspiciousness, matching human intuition (A >B >C) ? 176

11.3 **ZOORANK is effective.** (a) It gives nearly 100% accuracy while identifying suspicious users in the SWM dataset. (b) ZOORANK marks products reviewed by known fraudsters as suspicious. (c) Product #2 received nearly all of it's reviews by fraud users on one single day. 185

11.4 **ZOORANK is generalizable**. ZOORANK outperforms the baseline across different modes (see (a) and (b)) and across multiple datasets (see (c) and (d)) 186

11.5 **ZOORANK identifies fraudulent suspicious behavior in Twitter**: Top 100 suspicious users, and top 100 products as identified by ZOORANK. We can notice clearly the groups of suspicious users. 187

11.6 **Scalability of ZOORANK** (a) ZOORANK scales linearly with number of records. (b) ZOORANK scales linearly with number of blocks we want to find. 188

12.1 Illustration of OJRANK. Filled instances are true anomalies, unfilled are nominals, color depicts similarity. Upon each feedback, OJRANK re-ranks the instances, aiming to (a) push up similar True-Positives (red filled) & (b) 'mute' similar False-Positives (orange unfilled); (a) helps reduce expert effort, and both (a,b) increase true positive rate. 192

12.2 **OJRANK** outperforms simple as well as state-of-the-art baselines *significantly* for two metrics - *precision@b* and *expert effort*. (See §12.4 for details) 193

12.3 *precision@b* remains reasonably stable upon varying (left) δ and (right) k (2 input parameters to OJRANK). Each line corresponds to one of all 14+8 datasets in Table 12.2. 205
12.4 Number of anomalies shown by each method over feedback rounds for several BENCHMARK DATASETS. 206
12.5 Number of anomalies shown by each method over feedback rounds for several CLUSTERED DATASETS. 207
12.6 Avg. runtime per update on several (left) BENCHMARK & (right) CLUSTERED datasets. OJRANK's response time is less than one fifth of a second, with low variance. 208

LIST OF TABLES

1.1 Overview of Thesis Structure 8
1.2 List of papers in this dissertation 9

3.1 Similar terms to firestorms in literature. 18
3.2 Firestorms considered in the literature. 19
3.3 Top 20 firestorm events from Feb, 2011 to September, 2014, sorted by the number of tweets. 26

4.1 Block groups from which the most users have sent geotagged tweets. 40
4.2 Selected Values of Moran's I in residuals 45
4.3 Spatial errors basic model, binary Rook contiguity 46
4.4 Spatial errors full model, binary Rook contiguity, users with >5 tweets only. . . . 47

5.1 Country-wise number of selfie casualties 55
5.2 Descriptive statistics of Dataset collected for Selfies 57
5.3 Location-based, Image-based and Text-based features used for classification of selfies. 63
5.4 Reasons marked by annotators for a selfie being dangerous. 64
5.5 Average accuracy (with standard deviation) for 10-fold cross validation over different classification techniques and different feature configurations for the down sampled dataset. 65

6.1 A sub-sample of the cities selected for analysis. 73
6.2 Brief description of the data collected. 73
6.3 Performance of various classification methods, using different base architectures on our ground truth dataset. 74
6.4 Performance of models on held-out set. 77
6.5 List of dependent variables used to estimate the number of driving snaps posted. . 83
6.6 Regression models for number of distracted driving snaps (N=130). 83

7.1 Vocabulary at different experience levels. 91

7.2 Salient words for two facets at five experience levels in movie reviews. 93
7.3 Dataset statistics. 102
7.4 MSE comparison of our model versus baselines. 103
7.5 Experience-based facet words for the *illustrative* beer facet *taste*. 105
7.6 Distribution of users at different experience levels. 105
7.7 Salient words for the *illustrative* NewsTrust topic *US Election* at different experience levels. 108
7.8 Performance on identifying experienced users. 108

8.1 Honeypot account summaries 116
8.2 Egonet summary statistics 121
8.3 Boomerang network summary statistics 122
8.4 Fraudster account reuse habits 123
8.5 Collusion between fraud providers 125
8.6 Per-service entropy (in bits) over account attribute distributions. 125

9.1 Dataset summary 136
9.2 % of instances where proposed models outperforms alternatives (higher is better, >50% implies superior performance). 141
9.3 Pearson correlation coefficients between parameters in original and simulated data. 149
9.4 MMD test statistics between original data and model-simulated data (lower is better). 150
9.5 Anomaly detection performance (AUROC) under various anomaly contamination %ages (higher is better). 152

10.1 Dataset Statistics 159
10.2 Precision and Recall for SHERLOCK, SSC and SynchroTrap on real data. 166
10.3 F1 score of SHERLOCK across different classification and attack models (Stage I). 167

11.1 Comparison of other methods and their features 178
11.2 Symbols and Definitions 179
11.3 ZOORANK is generalizable over multiple datasets, and multiple modes that exist in the dataset. 185

12.1 Qualitative comparison between OJRANK and related methods. 195
12.2 Summary statistics for two sets of data used in experiments: (left) BENCHMARK and (right) CLUSTERED. 201
12.3 ***precision@b* on BENCHMARK DATASETS.** Per dataset rank provided in parentheses (the lower the better). Average rank across datasets given in the last row. Symbols ▲ and △ denote the cases where OJRANK is significantly better than the baseline w.r.t. the Wilcoxon signed rank test, respectively at (p<0.01) and (p<0.05). 202

12.4 ***precision@b* on clustered datasets.** Per dataset rank provided in parentheses (lower is better). Average rank provided in the last row. Symbol ▲ denote the cases where OJRank is significantly better than the corresponding baseline w.r.t. the Wilcoxon signed rank test at ($p<0.01$). 202

12.5 List of clustered datasets. We list the type of instances and from what class were they sampled. 203

12.6 ***Expert effort* on benchmark datasets.** Per dataset rank shown in parentheses (lower is better). Average rank is in the second last row (*effort* in $\mathcal{S}$ space). Average rank for *effort* in $\mathcal{X}$ space also given in last row. Symbols ▲($p<0.01$) and △($p<0.05$) denote the cases where OJRank is significantly better than the baseline w.r.t. Wilcoxon signed rank test. 204

12.7 ***Expert effort* on clustered datasets.** Per dataset rank provided in parentheses (lower is better). Average rank is in the second last row. Average rank for *effort* in $\mathcal{X}$ space also given in last row. Symbols ▲($p<0.01$), △($p<0.05$) and ▽($p<0.1$) denote the cases where OJRank is significantly better than the baseline w.r.t. Wilcoxon signed rank test. 204

Part I

Introduction and Background

CHAPTER 1

INTRODUCTION

Online social networking platforms (e.g. Facebook, Twitter, Snapchat, Yelp) provide means for users to express themselves, by posting content in the form of images and videos. These platforms also allow users to interact with content (liking, commenting), other users (social connections, chatting) and also the items (through ratings, and reviews). These social systems, besides pumping in billions of dollars of value to the economy, has also affected contemporary modern society. The importance of such systems can be seen in various spheres of human life such as political, economic and social. All such social systems store each interaction occurring on their platform, including all types of users' interaction with any other element (users, items, ratings, comments, etc.) on the platform. Storing these interactions provides us with huge and rich data, which before existence of these platforms was rarely available.

Such type of rich data is not only limited to social networking systems, but exists on other platforms as well. For example, even in healthcare systems, a patient's medical diagnosis, drug administered, etc. are stored and could be leveraged for developing insights into epidemics or disease trajectory patterns. Another example is software development platforms, where multiple users might collaborate with each other by posting their software code to repositories, and insights derived from data could be used to make the platform more efficient for collaborations. Other potential interesting domains where such platforms which capture rich data might exist include financial domain (Robinhood, Venmo), e-commerce systems (Amazon, iTunes, Google Play), livestreaming services (Twitch, YouTube Live), taxi-sharing systems (Uber, Lyft), vacation rental services (AirBnB) and many others.

Availability of such rich data at large scale for platforms that are governing key aspects of human life provides us a unique opportunity for identifying various previously unexplored and useful patterns. One key application of exploring interesting patterns in this data lies in the field of **computational social science**. The existing sociological theories were developed over a small sample of humans and it is uncertain as to how they scale given such rich longitudinal data for different platforms. The availability of the interaction data allow us to argue about human behavior through the lens of these omnipresent social and technical systems. Mining useful and interesting patterns is also crucial to the platform owners as well. Their goal is to ensure that users remain engaged and keep using the system. Discovery based on how users are using the system can allow the platform owners to redesign and introduce features which can increase longevity of users on the platform.

A crucial application for discovering patterns of how users use the given systems is that it allows us to generate models for normative behavior, which in turn could be leveraged to create **anomaly detection** models. These models could be used to identify instances of fraudulent behavior on these platforms, thus ensuring that systems are working and being used as they were intended to be.

My thesis focuses on the following questions, all of which are fundamental to understand and improve the use of such large social platforms:

- **Q1. Characterizing User Behavior**: How can we model user behavior on social media platforms?
- **Q2. Anomaly Detection**: How can we differentiate the genuine user behavior from the deviant user behavior? and further, identify suspicious/malicious actors?

These questions are closely related and are two faces of the same coin. Modeling user behavior on social networks helps in understanding the normal usage patterns on these platforms, which are crucial to identify the deviation from this behavior, and hence identify suspicious actors exhibiting anomalous behavior.

1.1 Overview and Contributions

1.1.1 Characterizing User Behavior

Large scale online platforms, such as Facebook, Twitter, Github provides us with an unprecedented opportunity to study user behavior on these platforms at this scale. The rise of such platforms and growth in their membership offers us not only breadth of the data that we can analyze (i.e. multiple different interactions) but also depth (i.e. analyzing the interactions over a long time). As mentioned earlier, such rich data can be leveraged for both (i) testing and verifying previously limitedly tested sociological theories, and (ii) provide redesign recommendations and interventions to the platform owners. For this part of my thesis, I study various social and recommendation platforms, including Twitter, BeerAdvocate, RateBeer, Snapchat. Further, we segment this part of thesis into two parts. The first part deals with topics related to the dynamics of user-based phenomenon, where we concentrate on how users collectively are part of a certain phenomenon on online social platforms. For first part, we characterize such observed user based phenomena on online platforms, and discover interesting patterns. In the second part, we concentrate not on the phenomena but on the individual entities that is the user, itself. For this part, we concentrate on behavior that is individual centric, and not the overall phenomena they might be exhibiting.

An interesting phenomena that exists on social media is firestorms i.e. a large amount of negative attention directed at a particular entity in relatively short time. This negative attention is generally due to some real-world event. In Chapter 1, we study if there is an impact of such firestorms. We ground this problem in a social science theory on biographical consequences of activism and study the mention networks formed during the firestorm. We were also interested in understanding the demographic distribution of geocoded users on Twitter. In Chapter 2, we conducted an analysis on US geotagged tweets, and correlated it with the demographic information obtained from census data.

Online social media platforms often provide users with a way to express themselves. However, in certain situations, users often ending up participating in dangerous risk-taking activities in offline world and post it online. We studied this behavior and presented models for identifying such posts on social media. First, we focussed on characterizing and detecting dangerous selfies in chapter 3. In Chapter 4, we concentrate on detecting distracted driving posts on social networks. We propose multiple classifiers to detect both such phenomena. We use the proposed classifiers to detect such dangerous content and then discover interesting insights about the entire phenomena. Most of the recommender systems capture user preferences and item facets, and some recent models also capture the temporal nature. However, one thing that has not been accounted for is the individual expertise of the users. An individual's taste might develop, and hence their preferences towards certain items might change. In Chapter 5, we propose a recommender system that captures individual expertise into account, while recommending products.

Contributions: In [160], we study and model the change in conversation network of participants when they participate in a firestorm on Twitter. We discover that the mention network before and after the firestorm are very similar to each other and very distinct from the week of the firestorm. We further corroborate this observation via means of testing network level statistics. Using sociological theory on *biographical consequences of activism*, we conclude that the firestorms on Twitter does not have a lasting effect. In [188], we study how does demographics of a certain region affect the amount of geospatial data created on Twitter and how can demographics be used to estimate this number for every census tract in the US. We discover that our model involving various census demographic variables explains 40% of the variance. In [161], we analyze the users who participate in voluntary risk-taking activities such as clicking dangerous selfies, exposing themselves to potential physical harm. We propose multimodal deep-learning models to predict if a particular selfie is dangerous or not, and achieve upto 82% accuracy. Continuing the voluntary risk-taking activities, in [164], we propose a deep learning classifier to identify distracted driving videos posted on online social media platform. We are able to achieve 92% accuracy and identify the distracted driving content posted over entire month of data. We use this data to argue about the cities in which such behavior is predominant. Additionally, we also model the individual users and how do they gain experience in review websites using their ratings and review text and use it to recommend products to users taking into account their expertise [209]. We achieved mean square error of 0.309 on RateBeer dataset, outperforming other baseline algorithms. Similar results are obtained for other domain datasets such as BeerAdvocate, NewsTrust, Amazon and Yelp.

Impact:

1. [160] won the Best Student Paper award at ASONAM, 2015.
2. [161] has been covered by more than 100 media outlets.
3. [161] has also been mentioned in numerous talks and educational programs conducted for high schools in India, and also provided content for a TEDx talk on the same topic.
4. Prior work to [188], [159] done by same authors, was runner up for SBP Data Challenge.

5. [188] is one of the top cited papers in ICWSM 2015, having >80 citations.[1]

1.1.2 Anomaly Detection

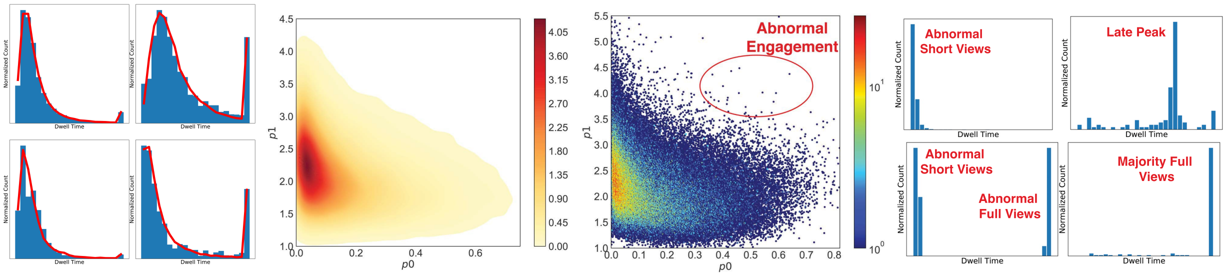
Recently, these systems have been a target of abuse and fraud, which have resulted in major societal implications. For example the growth in the number of fake reviews, fake news, fake accounts has had implications on both the political and commercial facets of society [152]. Detecting such fraudulent activities has become is crucial for the society as well because of it's widespread effect on its workings. It is also equally important for web platforms to get rid of the fraudulent activities on their websites to ensure that they work as they were intended to and assure users of the trustworthiness of the platforms.

Fraud on Twitter has been widely studied, with special emphasis on catching fake followers. However, there exist underground services that provide fake followers to paying customers. In Chapter 6, we study multiple fake follower services and characterize the type of fake followers in terms of their attributes they provide. Another similar type of market exist for providing fake views to multimedia content. In Chapter 7, we study dwell time engagement patterns of views. We propose parametric, interpretable models that not only help us understand normative engagement behaviors but also identify fraudulent engagement. In Chapter 8, we study fraudulent behavior in chatrooms of livestreams. We propose two-stage model to first identify chatbotted livestreams, and then subsequently detect chatbots participating in those livestreams.

In Part V, we propose methods that are not specific to social network data but can be extended to other settings. The popular methods for detecting lock-step suspicious behavior outputs a block of users and their corresponding activities that are considered suspicious. However, none of these methods provide a way to individually rank these entities based on their participation in such lock-step behavior. In Chapter 9, we present a method to do this. We further also showcase how different temporal features can be generated and used in fraud detection methods. An important question is what to do with the ranked list of suspicious entities that is often provided by any fraud detection algorithm. In Chapter 10, we propose a framework, where expert goes through the list of anomalies and provides feedback. Our framework, uses this feedback to rerank anomalies increasing precision and decreasing expert's effort for providing feedback.

Contributions: In the second part, we propose methods for identifying and detecting malicious actors on social media platforms. Further, we also propose anomaly detection models which are not confined to such platforms and can also be used for any general data. We analyze the underground Twitter follower market, discovering different types of link-fraud, and proposed "hard to game" features to successfully detect customers of such market [255]. The proposed method outperforms other baseline feature models with 0.98 precision and 0.95 recall. Continuing the work on detecting fraudulent engagement, we propose interpretable models for dwell time engagement on multimedia content posted on online social media platforms, and then use those models to identify fraudulent engagement [158]. The entire end to end methodology is shown in Figure 1.1, where we first propose individual parametric and interpretable models for characterizing each piece of content posted. In the next step, using copula fusion approach, we jointly model the parameters obtained from individual modeling of all the content samples in our

[1] As reported by Google Scholar on Nov 1, 2019



(a) Individual models (b) Group modeling (c) Anomaly detection (d) Dwell time anomalies

Figure 1.1: Proposed method [158] proposes (a) state-of-the-art parametric models for individual sample dwell times for content which closely mirror empirical data, (b) flexible copula modeling of aggregated multivariate parameter fits, (c) utilization of aggregate models for detecting dwell time engagement anomalies which (d) reflect abnormal behaviors radically inconsistent with most samples.

dataset. Finally, we use the joint model to identify anomalies, some of which we show in the rightmost panel of the figure. We show that the proposed models are robust to synthetic attacks of various types achieving high AUC greater than 0.9 and was able to successfully detect notable cases of fraud.

In [127], we work on the problem of detecting chatbots on livestreaming platforms. We propose a two stage detection mechanism where we first detect chatbotted streams and then work on detecting constituent chatbots in those streams. Our proposed method has a 97.4% precision over other baseline methods. For methods extending beyond social networks, we specifically analyzed group-level fraudulent activities, where users often collaborate with each other to attack machines, products (bringing the rating down), act as fake followers for an account. We proposed an individual level scoring mechanism to score each actor based on their participation levels in multiple such attacks [163]. We showed near perfect precision and recall over multiple datasets. In [157], we proposed a framework that takes into account human feedback to improve anomaly detection algorithms, while decreasing the human effort to provide feedback. We have also proposed models for identifying insider threats based on modeling activity sequences on user interaction with underlying software architectures [162]. We were able to outperform the existing baseline methods in both reducing false positive rate and also reduce the expert effort of labeling.

Impact:

1. [157] won the Best Research Paper award at SDM, 2019
2. [157] has featured in KDD 2019 tutorial on rare category exploration.
3. [163] has been downloaded 1.9K times.
4. [163] has been mentioned in keynotes at HotSOS 2016

1.2 Thesis Organization

This document is structured as follows: In Chapter 2, we provide the basic background information for commonly used ideas and techniques in this thesis. In the next 2 parts (II,III), I present my work related to user behavior characterization on social media platforms). Following which, I present my work on anomaly detection (IV,V). Finally I present conclusions and directions of future work in Part VI.

On social media platforms, users generally participate in a movement or a phenomena collectively. We try to characterize such user-based phenomena, where we are interested not in the individual users but the collective behavioral patterns observed. The user-based phenomena on social media is presented in Part II. This is different from the cases where it is more relevant to model individual users themselves, and understand their behavior. We study and model individual user behavior and present it in Part III.

In this thesis, we propose models and algorithms that detect fraudulent or anomalous activity on social media platforms. For these methods, we leverage a certain type of activity or features that are characteristic of the social media platforms, or the social media domain in general. We present this in Part IV. However, we also propose algorithms that can be applied to multiple domains, and not just necessarily social media domain. We present such methods in Part V.

Table 1.1: Overview of Thesis Structure

Modeling User Behavior on Social Technical Systems			
Understanding User Behavior		Anomaly Detection	
User based Phenomenon (Part II)	Individual User Modeling (Part III)	Fraud in Social Media (Part IV)	Beyond Social Media (Part V)
Ch. 1 Understanding firestorms [160] Ch. 2 Measuring bias in geocoded tweets [188]	Ch. 3 Detecting dangerous selfies [161] Ch. 4 Detecting distracted driving behavior [164] Ch. 5 Modeling user experience for recommendation [209]	Ch. 6 Faces of Link Fraud [255] Ch. 7 Modeling dwell time fraud engagement [158] Ch. 8 Detecting chatbots on Livestreaming platform [127]	Ch. 9 Individual Scoring in Group Fraud [163] Ch. 10 Incorporating Human Feedback in Anomaly Detection [157]

An overview of papers covered in this dissertation is shown in Table 1.2.

Table 1.2: List of papers in this dissertation

Reference	Title	Venue
[160]	A Tempest in a Teacup? Analyzing Firestorms on Twitter	IEEE/ACM International Conference on Advances in Social Network and Mining (ASONAM), 2016
[188]	Population Bias in Geotagged Tweets	International AAAI Conference on Web and Social Media (ICWSM), 2015
[161]	From Camera to Deathbed: Understanding Dangerous Selfies on Social Media	International Conference on Web and Social Media (ICWSM), 2017
[164]	Driving the Last Mile: Characterizing and Understanding Distracted Driving Posts on Social Networks	International Conference on Web and Social Media (ICWSM), 2020
[209]	Experience-aware Item Recommendation in Evolving Review Communities	International Conference on Data Mining (ICDM), 2015
[255]	The Many Faces of Link Fraud	International Conference on Data Mining (ICDM), 2017
[158]	Modeling Dwell Time Engagement on Visual Multimedia	SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2019
[127]	Characterizing and Detecting Livestreaming Chatbots	IEEE/ACM International Conference on Advances in Social Network and Mining (ASONAM), 2019
[163]	zooRank: Ranking Suspicious Entities in Time-Evolving Tensors	European Conference on Machine Leaning and Principles of Knowledge Discovery in Databases (ECML-PKDD), 2017
[157]	Learning On-the-Job to Re-rank Anomalies from Top-1 Feedback	SIAM International Conference on Data Mining (SDM), 2019

CHAPTER 2

PRELIMINARIES AND BACKGROUND

We begin with an overview of platforms, notations and concepts that we will use through the entire document. We will provide further necessary details, if needed, in each chapter.

2.1 Online Social-Technical Platforms

The thesis is focussed on characterizing and modeling user behavior on social and technical online platforms. Hence, we start by defining the various social media platforms we use throughout the thesis. We first focus on specific social networking platforms, and then give brief description of various other type of platforms.

Online Platforms: For this document, we define an *online platform* is a online website where users can interact with other users, or other elements (often representing another entity in offline world).

Facebook: Facebook is an online social networking platform, where users connect with other users forming friendship links. Users can post text, images and videos, which can be liked or commented on by other users on the plaform. Further, Facebook also has pages (generally referring to an organization) and groups, which can also create and post content.

Twitter: Twitter is a microblogging service, which is primarily used by it's users to update their peers through character limited text and images. Twitter is arguably also used to receive news [153], and follow high-profile celebrities. Twitter allows for directed relationships i.e. follower/followee relationship implying that it is not necessary for you following a user means the user also follows you back.

Twitch: Twitch is a livestreaming social platform, which is used to watch and stream broadcasts.Twitch is generally used by users to stream their gameplay or their lives, or lectures. Besides the broadcast, Twitch also allows the viewers to engage in conversation over the stream content through their chatstreams.

Snapchat: Snapchat is a messaging application that is used to share photos, videos and text. The major feature of Snapchat is that the content shared on the platform is ephemeral i.e. it will automatically get deleted after viewing or user defined timelimit. Snapchat also allows for directed relationships i.e. followers/ followee relationships. Besides messaging, Snapchat also has a map-based feature called SnapMaps, where users can anonymously share their content with location information, which appears on the map accessible by all users on the platform.

E-commerce systems: An e-commerce system (electronic commerce) allows users to sell or buy products and services using a web based platform or a mobile application. These systems generally allow users to also rate or review their purchases or products through text, images and ratings. Examples of such system includes Amazon, Flipkart, Software Marketplace.

Online Rating system: An online rating system is very similar to e-commerce system, major difference being that the platform itself does not allow for purchasing or selling of products. Examples of such system include BeerAdvocate, RateBeer, Netflix, and Yelp.

2.2 Graphs

One of the key data structure used prominently in the thesis is a graph. A graph is broadly defined as a set of nodes (also called as vertices) $\mathcal{V}$ and a set of edges $\mathcal{E}$ that connect nodes. In our examples, nodes are often used to represent users or items, and edges are between different users, or users and items and denote interactions in various forms.

Undirected graphs: The undirected graphs are used to define symmetric relationship between nodes. An edge exists between $u \in \mathcal{V}$ and $v \in \mathcal{V}$ if $(u, v) \in \mathcal{E}$ or $(v, u) \in \mathcal{E}$. Examples include friendship graph on Facebook, where the connection between nodes imply friendship and edge exists only if both nodes agree of the friendship.

Bipartite graphs: A bipartite graph is a graph that is use to map the relations between two disjoint and independent set of vertices $\mathcal{U}$ and $\mathcal{W}$. We modify the previous graph notation in case of a bipartite graph as follows: $\mathcal{G} = (\mathcal{U}, \mathcal{W}, \mathcal{E})$. The two disjoint sets of nodes $\mathcal{U}$ and $\mathcal{W}$ are generally nodes of different classes, i.e. $\mathcal{V} = \mathcal{U} \cup \mathcal{W}$ and $\mathcal{U} \cap \mathcal{W} = \emptyset$. As such edges in a bipartite graph can be represented as, $\mathcal{E} = \{(u, v) \mid u \in \mathcal{U}, v \in \mathcal{W}\}$. Such bipartite graphs can be used to represent multiple online interactions such as users buying products on e-commerce websites such as Amazon, users liking pages on social media websites on social platforms like Facebook, users reviewing restaurants on reviewing services such as Yelp.

Directed graphs: A directed graph is used to represent data in applications where relationship between nodes is not symmetric. For example, on Twitter, following relation is non-symmetric i.e. user A follows user B but that does not mean that user B does not follow user A. Edges in a directed graph, unliked undirected graphs, are an ordered pair of vertices, where $(u, v) \in \mathcal{E}$ does not imply that $(v, u) \in \mathcal{E}$.

Subgraphs: A subgraph is only part of a graph that is defined over a subset of nodes and edges in the entire graph. Given a subset $\mathcal{V}' \in \mathcal{V}$, we define the induced subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}') \subseteq \mathcal{G}$ where $\mathcal{E}' = \{(u, v) \in \mathcal{E} \mid u, v \in \mathcal{V}'\}$. A subgraph is a graph which contains the all nodes in the subset $\mathcal{V}'$ and all edges between these nodes.

Graphs as matrices: Graphs can be represented in the form of adjacency matrices. A unipartite graph $\mathcal{G}$ can be represented by $\mathbf{X} \in \mathbb{R}^{n \times n}$ where n is the number of vertices in $\mathcal{G}$:

$$\mathbf{X}_{u,v} = \begin{cases} 1, & \text{if } (u, v) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

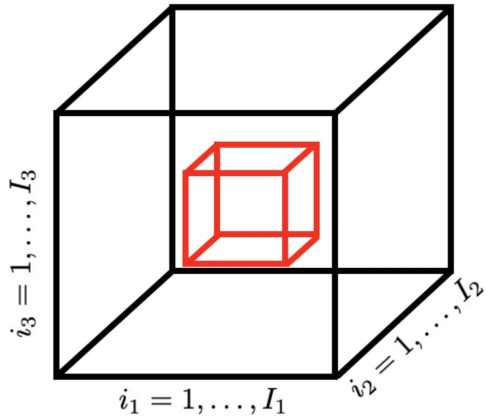


Figure 2.1: Illustration of a 3-mode Tensor $\boldsymbol{\mathcal{X}}$ having dimensions $I_1 \times I_2 \times I_3$, with a sub-tensor $\boldsymbol{\mathcal{Y}}$.

A graph will be undirected if $\mathbf{X}_{u,v} = \mathbf{X}_{v,u} \forall \{(u,v) \mid u \in \mathcal{V}, v \in \mathcal{V}\}$.

Similarly, a bipartite graph, $\mathcal{G} = \{\mathcal{U}, \mathcal{W}, \mathcal{E}\}$ can be represented by $\mathbf{X} \in \mathbb{R}^{n \times m}$ where $|\mathcal{U}| = n$ and $|\mathcal{W}| = m$.

$$\mathbf{X}_{u,v} = \begin{cases} c((u,v)), & \text{if } (u,v) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

where $c((u,v))$ capture certain descriptive property (ratings, strength, count) of the edge.

2.3 Tensors

Another mathematical structure, we use extensively is tensor. We here discuss the basic definition and notations related to tensors.

Tensor: A *tensor* is a multi-dimensional array of entries. The *order* is defined as the number of dimensions, also called *modes*. Consider an N-order (or N-way) tensor $\boldsymbol{\mathcal{X}}$ of size $I_1 \times I_2 \times \cdots \times I_n$. Each entry indexed by $(i_1, \dots, i_n)$ can be denoted by $x_{i_1,\dots,i_n}$. Each index i_n runs from 1 to the maximum length of the mode i.e. I_n. I_n is also known as the dimensionality of mode n. Figure 2.1 shows an illustration of a tensor. A tensor is a useful mathematical tool used to represent multidimensional data efficiently.

Sub-Tensor: A N-order subtensor $\boldsymbol{\mathcal{Y}}$ of N-order tensor $\boldsymbol{\mathcal{X}}$ is obtained by removing certain slices from $\boldsymbol{\mathcal{X}}$.

Examples: To explain the applicability of tensor, we give certain ways a tensor can be used to formulate interesting concepts in various domains.

- **Social Network Friendships:** A 3-order tensor $\boldsymbol{\mathcal{X}}$ of the form *user*×*user*×*date* can be used to represent as to on what date two users became friends. Each cell of the tensor has the value x_{i_1,i_2,i_3}, which is set to 1 if i_1-th user became friends with i_2-th user on i_3-th date.
- **E-Commerce:** A 3-order tensor $\boldsymbol{\mathcal{X}}$ of the form *user*×*product*×*date* can be used to represent on what date does a user buys/reviews/rates a product. Each cell of the tensor has the

value x_{i_1,i_2,i_3}, which is set to either the number bought or rating given of the product i_2 by the user i_1 at date i_3.

- **Network Traffic:** A 3-order tensor $\boldsymbol{\mathcal{X}}$ of the form *sourceIP*$\times$*destIP*$\times$*port* can be used to represent the number of connections from source IP address to destination IP address on a specific port. A single entry x_{i_1,i_2,i_3} can be used to represent the total number of instances when source IP i_1 sent a packet to destination IP i_2 on port i_3.

2.4 Learning

The thesis also has couple of chapters that are focused on learning from the given data. We now give a high-level overview of generalized learning framework that we use in this document.

Objective Functions: We generally use models to approximate the given data through a data generating process, characterized by certain model parameters. The goal is to be able to identify model parameters that allow us to approximate the observed data well. This "goal" is mathematically expressed through an objective function, which generally captures a form of relationship of closeness between the approximated and observed data. For example, SVD minimizes the Frobenius norm between our data and the model:

$$\arg\min_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\|_F^2$$

Regularization and Constraints: In many cases guided by the application domain, we might want to include constraints and regularizations. A popular form of regularization is to introduce *sparsity* in the learnt model parameters. This is achieved by adding $l1$ norm over the parameters to the objective function.

Optimization: Given an objective function, the next step is to optimize or efficiently learn parameters that minimize (or maximize, based on how it is defined) the given objective. Many different optimization algorithms exist [271] and are generally chosen on the basis of meeting the requirements of optimization problem, efficiency and convergence guarantees. With the advent of deep-learning, stochastic gradient descent [31] is a really popular optimization algorithm. We will discuss more of the optimization algorithms in depth in the relevant chapters.

Part II

User-based Phenomena on Social Media

Hemank Lamba, Momin M. Malik, Juergen Pfeffer. ”A Tempest in a Teacup: Analyzing Firestorms on Twitter.” 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2015.

CHAPTER 3

MEASURING THE IMPACT OF FIRESTORMS

‘Firestorms,’ sudden bursts of negative attention in cases of controversy and outrage, are seemingly widespread on Twitter and are an increasing source of fascination and anxiety in the corporate, governmental, and public spheres. Using media mentions, we collect 80 candidate events from January 2011 to September 2014 that we would term ‘firestorms.’ Using data from the Twitter decahose (or gardenhose), a 10% random sample of all tweets, we describe the size and longevity of these firestorms. We take two firestorm exemplars, #myNYPD and #CancelColbert, as case studies to study them in detail. Then, taking the 20 firestorms with the most tweets, we look at the change in mention networks of participants over the course of the firestorm as one method of testing for possible impacts of firestorms. We find that the mention networks before and after the firestorms are more similar to each other than to those of the firestorms, suggesting that firestorms neither emerge from existing networks, nor do they result in lasting changes to social structure. To verify this, we randomly sample users and generate mention networks for baseline comparison, and find that the firestorms are not associated with a greater than random amount of change in mention networks.

On Twitter, firestorms (or Twitterstorms) have become an object of fascination and anxiety. They are one of the major topics in discussions of Twitter in the realm of public relations and brand management [221]. Individual events frequently receive media coverage, and the phenomenon as a whole receives coverage as well; political comedian John Oliver did a segment critiquing corporations’ use of Twitter on the September 15, 2014 episode of his HBO television show, *Last Week Tonight*, featuring many examples of firestorms. Online magazine *Slate* dubbed 2014 the ‘Year of Outrage’ in an eponymous special feature, listing one example for every day of the year (each with an accompanying tweet to illustrate the outrage) alongside reflection articles such as ‘The Life Cycle of Outrage.’

The term ‘firestorm’ refers to an event where a person, group, or institution suddenly receives a large amount of negative attention [229]. Any sudden controversy or expression of outrage may be termed a firestorm, although we are interested in a firestorm as something more specific:

a case where the sudden negative attention is in response to a recent action or statement of the target entity (rather than without a specific trigger, such as in a premeditated protest or prank) and arises spontaneously (rather than through prior coordination, such as from a group prepared for mobilization). Furthermore, we are interested in when this attention exhibits network effects: the initial negative attention causes more people to learn of the action or statement, and these people then contribute their own negative attention. Such cases are examples of negative word-of-mouth dynamics. We focus on firestorms targeting public figures, businesses, and institutions, where consequences are public; we do not consider firestorms targeting private individuals [240], as the consequences there are in terms of the individuals' experiences which we consider a different topic.

The ultimate question is if participation in or consumption of firestorms has an effect outside of Twitter, such as through purchasing decisions, voting behavior, attendance at protests, or even participation in violence, either directly (by firestorm participants) or indirectly (by people influenced by firestorm participants). However, such information is impossible to collect directly at scale, and difficult even to indirectly infer from Twitter data. Instead, we draw on literature about the *biographical consequences of activism* [192] to ask, can we detect a change in firestorm participants as a result of the event? We look specifically at social ties of firestorm participants and form the research question: what is the relationship between social ties and firestorm participation? I.e., do the people who participate in a firestorm know each other beforehand? Do they communicate during? And do they continue to communicate after? If there is a discernible change in social ties over the course of a firestorm, it suggests a social impact that could lead to long-term consequences. On the other hand, if firestorms arise from existing social ties, it would point to firestorms being a consequence rather than a cause of other action, and if there is no relation to social ties, it would be inconclusive but, as social actions are embedded in networks of social ties, it would suggest firestorms are of little importance.

3.1 Background and Related Work

3.1.1 Firestorms

There have been several papers directly on Firestorms. We summarize these in tables 3.1 and 3.2.

NAME	ARTICLE(S)
Crises	Bruns & Stieglitz 2012 [37], Park et al. 2012 [224], Rajasekera 2010 [235]
Scandals	Bruns & Stieglitz 2012 [37]
Bad news	Park et al. 2012 [224]
Firestorms	Mochalova & Nanopoulos 2014 [202], Pfeffer et al. 2014 [229]
Shitstorms	Stieglitz & Krüger 2014 [266]

Table 3.1: Similar terms to firestorms in literature.

YEAR	FIRESTORM	ARTICLE(S)
2009	Domino's employees prank video	[224]
2010	Toyota recall	[68, 235, 265]
2011	Playstation Network hack and shutdown	[221]
2012	#QantasLuxury campaign after labor dispute	[37, 221, 229, 265]
2012	Papa Johns "lady chinky eyes" receipt	[224]

Table 3.2: Firestorms considered in the literature.

Much of this literature is about the problem specification, with conclusions being very preliminary. What has been found so far is that external events such as statements do affect the firestorm [221, 266], but that there is a time lag in the diffusion of an apology [224], and that a small number of users are responsible for the vast majority of the tweets [37] just as in Twitter activity in general [119].

Twitter's culture makes brands particularly vulnerable to firestorms. Van Dijck [284] discusses the "paradox of Twitter," one aspect of which is that the thing that gives Twitter value for marketers—the authenticity and openness of social interaction—is destroyed when marketers try to intervene to capitalize on that value. Nitins and Burgess [221] write that early on, brands saw social media as instant and free access to consumers around the world. But they failed to consider the culture of social media, importing their standard one-to-many communication models. They quickly found that consumers also now have the ability "to 'talk back' to companies—even very large global corporations—[and] to do so in public; they can share their pleasure, or displeasure, with potentially millions of other consumers without significant effort," and that they often resented the intrusion of companies. Because of this, they continue, "Twitter users frequently delight in 'gotcha' moments". Note that this scenario may be very different for certain celebrities and brands who develop a strategy of appealing to iconoclasm, for whom frequent negative attention may be beneficial and Twitter may be an easy way to garner this (e.g., potentially Kenneth Cole and Urban Outfitters).

In terms of modeling firestorms, there is relevant literature on 'media hypes' or 'media storms,' and on news cycles. Vasterman [286] characterizes media hypes as being *self-reinforcing*, potentially being driven less by external events after the triggering event and more by discussion about itself [286, 298] until the issue is crowded out by another topic. Vasterman suggests a smooth left-skewed distribution as a model, while Wein and Elmelund-Præstekær [298] find evidence of a decreasing oscillatory pattern. For news cycles, Leskovec et al. [172] found a 'saw-toothed' shaped increase followed by an exponential decay, which they were able to reproduce in a simulation model that combined an imitation effect and a recency effect.

It is often assumed that firestorms have an effect, and are therefore important. However, whether or not this is so is an open question. Kimmel and Kitchen [142] argue that while word-of-mouth, including negative word-of-mouth, is significant in shaping consumer attitudes and behavior, its power is also frequently oversold. They urge caution towards claims of the impact of word-of-mouth on social media. More generally, the question of whether low-commitment online protest and activism has an impact is hotly debated. Many argue that 'slacktivism' [203] or ‘clicktivism’ [297] are not effective at achieving their aims, while others argue that ‘hashtag activism’ [42] is better than nothing.

A more nuanced way of looking for an impact from firestorms is to consider the effects on participants themselves. McAdam’s famous work [192] introduced the idea that participating in activism has an impact on individuals, and even if the given activism itself is not successful, it has ‘biographical consequences.’ Individuals influenced by earlier participation go on to do further actions that are significant. Indeed, looking at activists rather than campaigns shows that online activism plays a role in larger movements even when specific campaigns have no impact [50], and this is important to consider when judging the effectiveness of online action [248].

In order to address our research question of the relationship between social ties and firestorm participation, we look at mention networks. Merritt et al. [197] have shown in another context that discussion is an effective proxy for friendship ties. We would go further to say that while we cannot measure exposure from friendship data as in Myers and Leskovec [212] from the available data, using mentions as the measure of social ties is a stronger mark of connection and thus a more meaningful measure. This is different from Granovetter’s [92] concept of a strong tie, but a mention is nonetheless a strong*er* tie than a following relationship and we would expect its impact to be greater than just a follower relationship.

3.1.2 Twitter

Amidst the enormous recent academic literature involving Twitter [299], researchers are increasingly beginning to appreciate that studying the microblogging platform is not necessarily the same as studying human behavior in general [245, 283]. There are multiple barriers to generalization, including that Twitter demographics are non-representative [187, 200]; that the possibility of making money from link farming [83] or from selling bots to inflate metrics [60] means there is widespread spam [277] that Twitter is not able to entirely or immediately filter out [276], and this spam may distort research findings [83]; that there are idiosyncratic conventions of Twitter [33, 129, 153] and a specific culture and ideology that is anti-establishment [221, 284] and, as shown by what drives adoption, focused on celebrity culture [110]; that even beyond spam, the Twitter social graph [77] has non-random patterns of adoption that potentially give it a topology vastly different from that of the underlying social network [249]; that the most accessible channel of data, the Streaming API, is unreliable within certain parameters [206], which causes difficulty in studying meso- and macro-scale phenomena [36]; and that Twitter is itself neither globally uniform [231] nor a static, stable environment across years [181, 284].

However, Twitter is host to many firestorms in itself. That is, there are frequent cases where a tweet sets off a firestorm, where an apology is given via tweet, or where a protest is organized under a hashtag. This, combined with how Twitter has become a critical channel of communicating and cultivating brand reputation and identity [150, 221, 266], means that firestorm behavior on Twitter is of interest in and of itself without needing to be representative of larger social behavior.

3.1.3 Specific firestorms

While we ultimately find 80 candidate examples of firestorms, we select two of them to examine more closely. The first is #CancelColbert, a hashtag started by activist Suey Park in reaction to a tweet quoting a skit on the satirical news program *The Colbert Report*, from American political comedian Stephen Colbert. The hashtag took off, and was soon followed by a reaction against the hashtag. We use it as an example of a firestorm that potentially comes from an already well-connected community, as initially it would only have been users following Park who would have seen her call to trend #CancelColbert.

Second is #myNYPD, a campaign started by the New York Police Department (@NYPDnews) to collect positive stories about the NYPD. However, it was 'hijacked' and used it as an opportunity to highlight grievances around police brutality: alongside sarcastic comments about the kindness of police, users posted pictures of NYPD officers grabbing, kicking, beating, and otherwise abusing people. The campaign was widely considered a failure and embarrassment for the NYPD, and is an excellent example of hashtag hijacking and public relations gone wrong.

3.2 Methodology

3.2.1 Firestorm Identification

As we note above, the link between activity on Twitter and larger societal phenomena is complex and difficult to disentangle from all the confounding factors. Thus, while people take to Twitter over practically every controversy or outrage, we decided that only firestorms that have some substantive connection to Twitter would be meaningful to study with Twitter data. We developed inclusion criteria, that a controversy first must have had some media mention, and second it must meet at least one of the following conditions:

- The controversy began around a tweet or series of tweets;
- The entity at the center of the controversy posts a apology, retraction, non-apology, or otherwise major statement on Twitter; or
- A specific hashtag, that we were able to find through searching through media, is associated with the controversy.

We also choose to exclude cases where it is obvious that something sent from a professional account was meant to be posted from a personal account, as we find that in such cases Twitter users are generally more amused than angry. We include cases of social media account managers failing to use proper discretion (i.e., intending to post what they did, but being mistaken that it was appropriate).

We limit the period under consideration to the middle of September 2014, as per our data. We only found several firestorms in 2009 and 2010, and no earlier examples, so we choose to consider only firestorms in 2011 onwards.

3.2.2 Firestorms collection

Our search method consisted first of web searches for "Firestorms" and Twitter, "shitstorms" and Twitter, and other variations; we went through any lists of "social media fails" we came across; and lastly, searched through tags like "Twitter" and "PR" on technology- and culture-oriented blogs and aggregation sites such as Buzzfeed, Mashable, and The Verge. We used other aggregated lists, such as on KnowYourMeme.com, John Oliver's segment, and the Slate feature on the 'Year of Outrage'. For each firestorm we collected the start date, the date of any apology or retraction if it exists, and all hashtags and handles associated with the firestorm (some firestorms are centered around a particular user, others around a hashtag).

3.2.3 Data Source

Our data source is an archive of the Twitter decahose, a random 10% sample of all tweets. This is a scaled up version of Twitter's Sample API, which gives a stream of a random 1% sample of all tweets. As found by Morstatter et al. [206], the Sample API (unlike the Streaming API) indeed gives an accurate representation of the relative frequencies of hashtags over time. We assume that the decahose has this property as well, with the significant benefit that it gives us more statistical power to estimate the true size of smaller events.

The decahose, like the Sample API, does not allow queries regarding the social graph, thus preventing us from modeling individual exposure to information [239]. And because information about when links were formed is not stored by Twitter, it is difficult to reconstruct the state of the social graph at a previous point in time [77]. However, following the demonstration in [197], we use mentions as a proxy for ties. We recognize that any given mention has only a 1/10 chance of being in our data, but this means that we are, on average, capturing ties that consist of at least 10 mentions. Since many of the mention networks we find are fairly dense, keeping only ties consisting of at least ten mentions would be justifiable even as a filtering strategy.

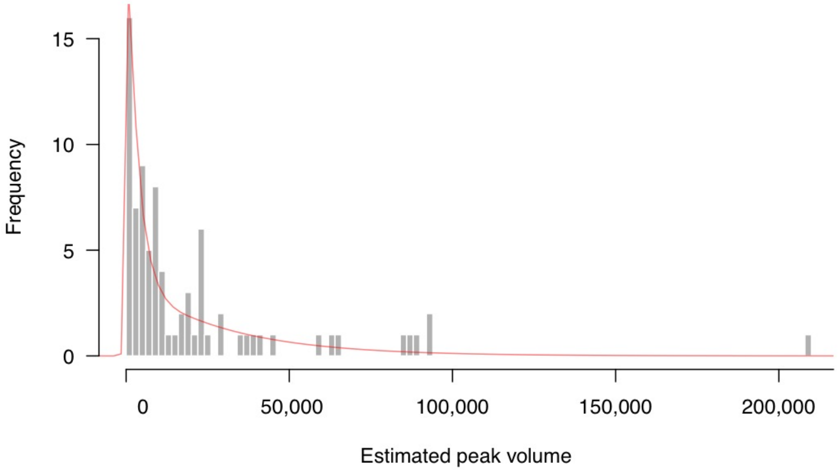


Figure 3.1: Histogram of peak sizes of collected firestorms, with a scaled fitted logspline. The x-axis is the estimated volume of tweets reached on the peak day, and the y-axis is the number of firestorms reaching that volume.

3.2.4 Data Extraction

We do pre-processing on the decahose data to simplify the computational task, extracting (1) daily summaries of co-present entities and the user who posted that tweet (e.g., if @user tweets

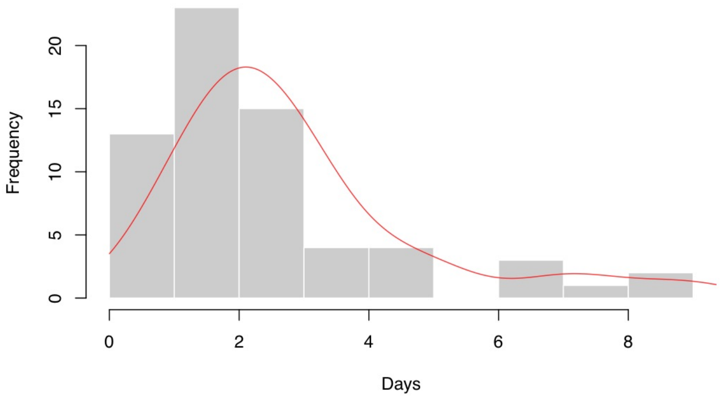


Figure 3.2: Histogram of distribution of the number of days it took the collected firestorms to decay to 90% of peak volume, with scaled fitted density.

“@alter #tag1 #tag2”, we would record the co-presence of @user, @alter, #tag1, and #tag2), and (2) daily tabulations of hashtag and mention frequencies for all such entities in the data by day. The aggregation by day is by the UTC timestamp in the tweets, which potentially splits firestorms across days as experienced by firestorm participants. Fine-grained extraction may be appropriate for future work. For each candidate hashtag or mention, we extracted the daily frequency from -5 days (for a baseline) to +60 days (for a tail) from the start date. We found that the tails died off well within 10 days, such that a smaller period would be sufficient for future extractions.

In addition to extracting frequency plots for all firestorms, and often for multiple entities (hashtags and mentions) for each firestorm to see if the firestorm was better captured by one entity or another, we extracted the full text and metadata of tweets for the 20 firestorms with the highest volume of tweets on their respective peak days. For these 20 we also constructed *mention networks* of all firestorm participants. This consisted of taking all usernames found in the firestorms (i.e., all uses who included in at least one tweet with the entity by which we identified the given firestorm) and extracting all mentions between them during the firestorm (including tweets not containing the firestorm entity). We did not consider mentions by or of users not participating in the firestorm. In order to do a pre- and post-firestorm comparison, we similarly collected all mentions between firestorm participants going back to two weeks before the firestorm, and forward to two weeks after the firestorm. We aggregated these into networks by one-week intervals. In order for mentions of the target’s Twitter handle (when there is a clear target) during the firestorm to not drown out other structure, we remove the node of the target handle during the firestorm week. For consistency, we remove the target handle from other weeks as well.

For spam filtering, we first did qualitative investigation of the data. Most of what we identified were tweets that contained a URL and a string of unrelated hashtags, e.g.,

> NEW F O L L O W E R S=&gt;[a URL here]
> #DescribeYourCrushIn3Words,#Brentto600k,
> #CancelColbert,#ULTRALIVE,#HowOldAreYou,Napier an

This led us to investigate a rule-based filtering system similar to that employed by Kwak et al.

[153]. However, as it turned out, spam tweets of this form accounted for less than half a percent of the total volume of tweets. Investigating the top hashtags for that day revealed no overlap, suggesting that the spam captured in our data was from spambots employing a minority strategy of tweeting out all currently trending topics. Because the volume of spam was negligible, and investigation showed the top tweeters in both of our case studies were indeed humans, we decided to not employ any filtering.

3.3 Results

3.3.1 Firestorms

For each of the 80 firestorms, we identified the one entity that best represents the firestorm, the number of tweets posted related to it in the first 7 days of the event, and the number of unique users who participated in this event. The day on which maximum activity in number of tweets was observed is referred to as the day of peak activity.

We found that most of our firestorms have an estimated peak volume of below 50,000[1] (fig. 3.1). The outlier with over 200,000 tweets is #WhyImVotingUKIP, although we didn't investigate why this might have been so large compared to other firestorms.

By comparing the date of the initial event to the date of the peak, we can see how quickly the firestorm reached peak activity; this was generally on the start date, for 64 out of the 80 cases. We can see from figure 3.3 that the larger firestorms in our collection do not take longer to decay, suggesting a phenomenon not tied to scale.

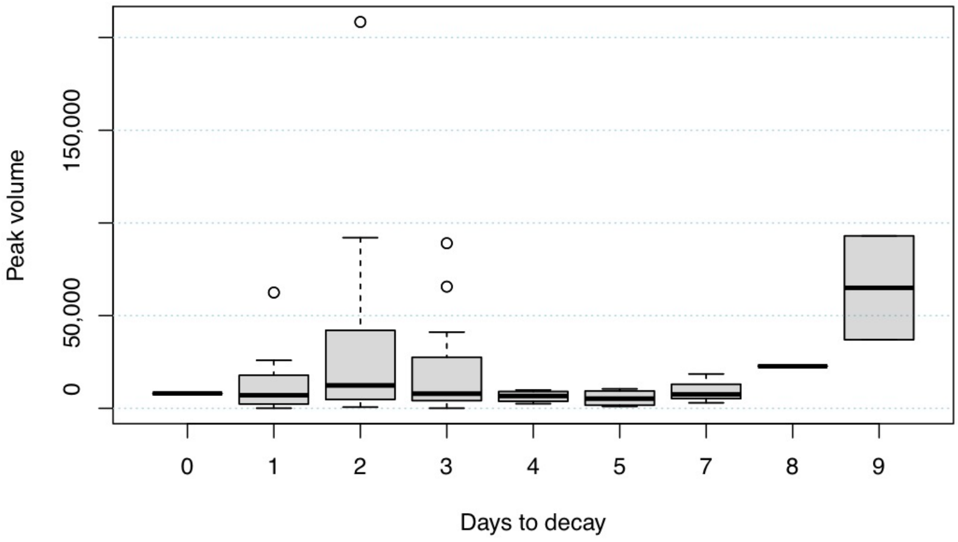


Figure 3.3: Boxplot of peak volume versus the days it took to decay to 90%.

As mentioned above, we focus on the 20 firestorms with the largest number of tweets on their peak day. Brief descriptions of each of the firestorms, along with their respective numbers

[1] The estimate is number of tweets observed from the decahose multiplied by 10 (sampling rate)

of tweets and dates, is given in Table 3.3. We excluded from the table and from fig. 3.2 any cases where the time to decay was more than 10 days. In such cases, the firestorm tweets did not exceed one-tenth of the average total volume of the given entity; in such cases, the firestorm likely would matter little to the target and may not even be noticeable, an indication that all things called firestorms are not necessarily meaningful to call as such.

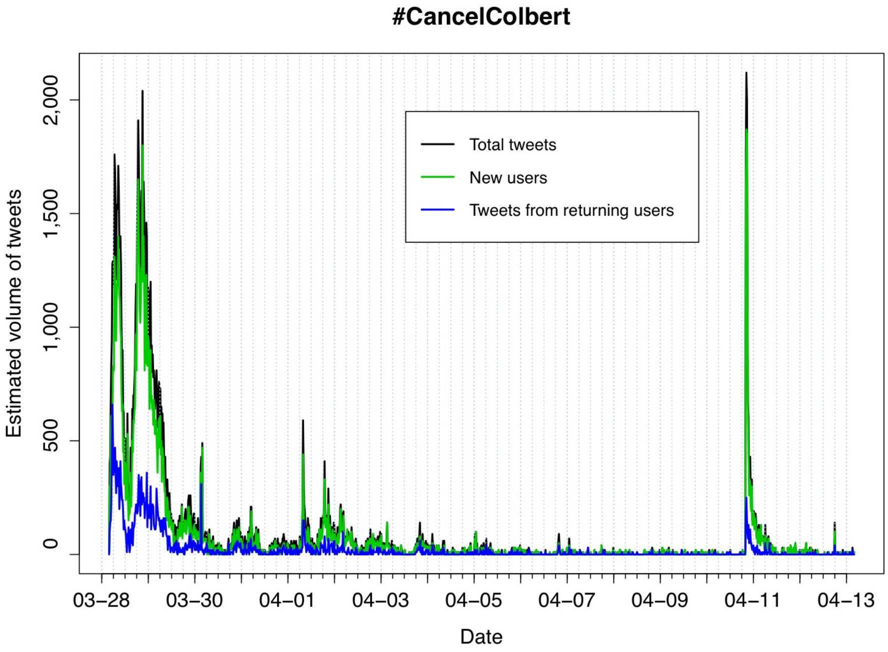


Figure 3.4: Estimated number of tweets with hashtag #myNYPD (case-insensitive) over 16 days in 2014, plotted at the midpoints of bins with a width of 16.67 minutes. Ticks are at 6 hour intervals, UTC. The number of tweets from users using the hashtag for the first time is used to measure the number of new users. Since the total number of tweets is very close to this number, we include the difference between the two, which are the tweets from returning users.

3.3.2 Case studies

The volumes of our two case studies, #CancelColbert and #myNYPD, are shown in figures 3.4 and 3.5 respectively. For these, we chose to show 16-day period because, after that, there was negligible activity (there was also very little activity in #myNYPD after the first week, but we included the same length of time for comparison). In both plots, the bimodal initial peak corresponds to the hours from late night to early morning.

Since the number of new users is almost identical to that of the total tweet frequency, we also provide a log-log plot of the distribution of tweets per user in figure 3.6 which shows a typical heavy-tailed distribution (here we provide raw decahose numbers as scaling by 10 would lose the head of the distribution, where most of the mass is).

For #CancelColbert, the first peak visible in the plot is the initial Twitter discussion after the tweet from @suey_park. Colbert discussed the campaign on his show on the night of March 31st

Firestorm Hashtag/Mention	Start Date	Tweets	Users	Source with description of the firestorm
#whyimvotingukip	2014-05-20	39,969	32,376	knowyourmeme.com/memes/events/whyimvotingukip
#muslimrage	2012-09-17	15,722	11,947	buzzfeed.com/ryanhatesthis/newsweeks-muslim-rage-cover-sparks-immediate-ba#.ljGJy1ykw
#CancelColbert	2014-03-27	13,277	10,349	newyorker.com/news/news-desk/the-campaign-to-cancel-colbert
#myNYPD	2014-04-22	12,762	10,362	knowyourmeme.com/memes/events/mynypd
#AskThicke	2014-06-30	11,763	9,699	knowyourmeme.com/memes/events/askthicke
@TheOnion	2013-02-24	9,959	8,802	hollywoodreporter.com/news/onion-calls-quvenzhane-wallis-c-424113
@KLM	2014-06-29	8,716	8,050	utsandiego.com/news/2014/jun/29/klm-adios-amigos-twitter-mexico/
#qantas	2011-10-26	8,649	5,402	reuters.com/article/2011/11/22/us-qantas-idUSTRE7AL0HB20111122
@David_Cameron	2014-03-05	7,096	6,447	theguardian.com/politics/2014/mar/06/celebrities-parody-camerons-on-the-phone-to-obama-selfie-tweet
@celebboutique	2012-07-20	6,679	6,189	mashable.com/2012/07/20/celebboutique-misguided-aurora-tweet-sparks-twitter-outrage/
@GaelGarciaB	2014-06-29	6,646	6,234	utsandiego.com/news/2014/jun/29/klm-adios-amigos-twitter-mexico/
#NotIntendedToBeAFactualStatement	2011-04-13	6,261	4,386	knowyourmeme.com/memes/events/not-intended-to-be-a-factual-statement
#AskJPM	2013-11-06	4,321	3,418	dealbook.nytimes.com/2013/11/13/after-twitter-fail-jpmorgan-calls-off-q-and-a/?_r=0
@SpaghettiOs	2013-12-06	2,890	2,704	huffingtonpost.com/2013/12/07/spaghettios-pearl-harbor-tweet_n_4404397.html
#McDStories	2012-01-18	2,374	1,993	businessinsider.com/mcdonalds-twitter-campaign-goes-horribly-wrong-mcdstories-2012-1
#AskBG	10-17-2013	2,221	1,933	bbc.com/news/business-24563421
#QantasLuxury	2011-11-22	2,098	1,657	reuters.com/article/2011/11/22/us-qantas-idUSTRE7AL0HB20111122
#VogueArticles	2014-09-10	1,894	1,819	washingtonpost.com/blogs/style-blog/wp/2014/09/10/vogues-celebration-of-big-butts-falls-flat-and-inspires-voguepitches/
@fafsa	2014-06-25	1,828	1,692	nbcnews.com/business/personal-finance/fafsa-im-poor-tweet-sparks-online-backlash-n140356
@UKinUSA	2014-08-24	142	140	washingtonpost.com/blogs/post-politics/wp/2014/08/25/british-embassy-apologizes-for-tweet-commemorating-the-burning-the-white-house-but-not-for-the-actual-burning-of-the-white-house/

Table 3.3: Top 20 firestorm events from Feb, 2011 to September, 2014, sorted by the number of tweets.

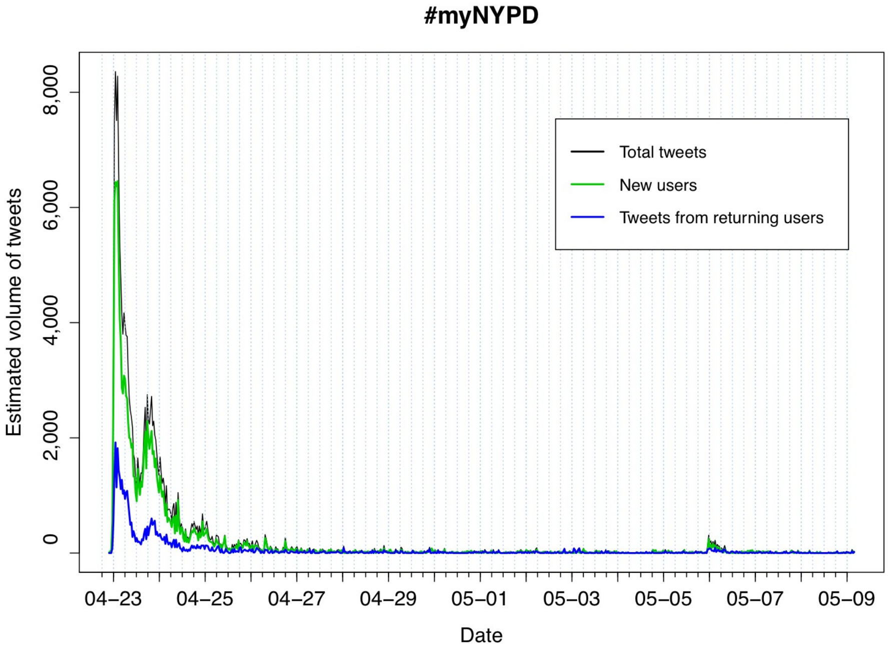


Figure 3.5: Estimated number of tweets with hashtag #CancelColbert (case-insensitive) over 16 days in 2014, with other details identical to that in fig. 3.4.

in a segment entitled, "Who's Attacking Me Now? - #CancelColbert", but this had little impact. The second peak is, interestingly, from April 10th, the day of a press release from CBS (later also covered on *The Colbert Report* that night) announcing that Colbert would be leaving his show to become the host of the famous American late-night show *The Late Show*. Much of the content of that spike were jokes about #CancelColbert having worked. The vast majority of these tweets were from users who had not participated in the initial firestorm (see fig. 3.4) and, further analyzing the tweets, there were almost no additional mentions of the users who were heavily mentioned before: @StephenAtHome has an estimated 14,190 mentions in the first 13 days and 1,610 in the next 13 days, and @suey_park has an estimated 11,970 mentions in the first period and only 980 in the second. This suggests far lower levels of interaction for the event that was not a firestorm than in the event that was, a topic for future exploration.

In #myNYPD, the users with the highest tweet volume appear to be members of the public, with the exception of @Copwatch, an activist network. The profiles with the highest indegree (most mentions) is revealing: while @NYPDnews is most mentioned, with an estimated 15,180 mentions (almost all in the initial 13-day period), second-most is @OccupyWallStreetNYC, with 10,860, followed by @YourAnonNews with 5,620, @Copwatch with 4,390 and @VICE with 3,580. The frequent mentions of Occupy show linking back to recent police action at Zuccotti Park against protestors from the Occupy Wall Street movement. The frequent mentions of @VICE are tweets linking to an article [2] about the firestorm quickly published on the website of

[2] "Disastrous #myNYPD Twitter Campaign Backfires Hilariously," https://news.vice.com/article/disastrous-mynypd-twitter-campaign-backfires-hilariously

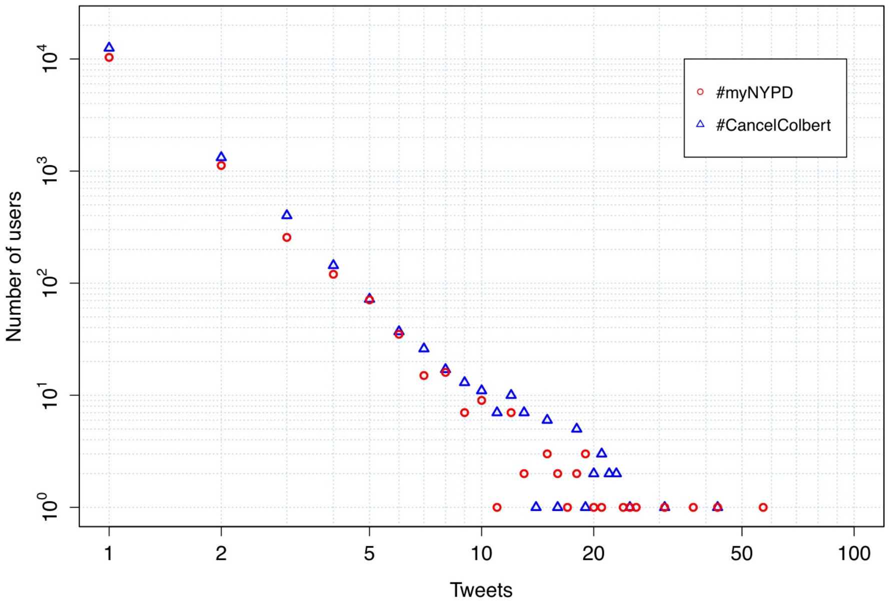


Figure 3.6: Distribution of tweets per user in the decahose (log-log scale).

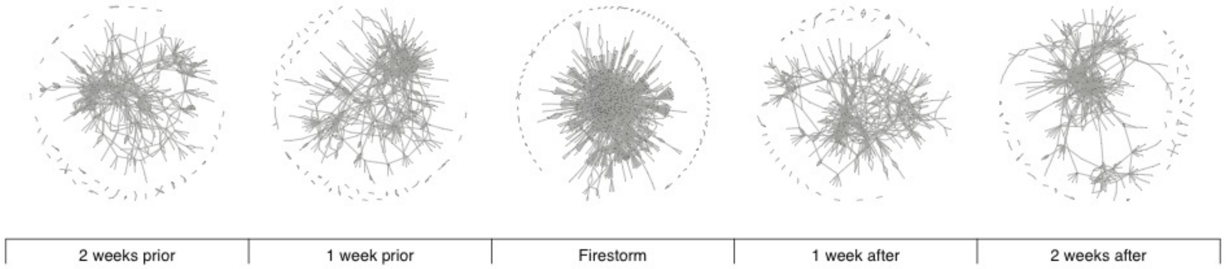


Figure 3.7: Network of mentions between firestorm participants, in this case for #askJPM, aggregated by week, before, during, and after the event.

Vice magazine, giving one possible example of a feedback loop between media and firestorms.

3.3.3 Mention networks

As discussed earlier, we create mention networks to study change in social interactions pre- and post-firestorm. When investigating the mention networks, we saw a characteristic pattern: the network of the week of the firestorm looked dramatically different from the others (fig. 3.7), with far more concentration and far less of a distributed network structure. Some of the normally present conversational structure seemed to disappear. We investigated a number of global network metrics (density, centralization, clustering coefficient/transitivity, reciprocity), but even when a given metric changed across networks from week to week, the change was not so great that 95% confidence intervals from week to week did not overlap. However, we found that there were far fewer edges in common between the firestorm week and the other weeks. We measured this formally with a Jaccard index (size of intersection divided by size of union) on the directed edges of the mention networks.

Figure 3.8 shows the distributions over the 20 top firestorms between each pair of weeks; we add vertical lines at the mode as it makes the difference more noticeable than lines at the means, but t-tests for comparisons of means still show that the difference in means between a non-firestorm week and a firestorm week is significant in all cases, and between any two non-firestorm weeks is non-significant. Surprisingly, even the pre-firestorm weeks and post-firestorm weeks were more similar to *each other* than to the firestorm, indicating that there is a minimum underlying social structure of discussion, relatively constant in time, but from which a firestorm departs.

The similarity between pre- and post-firestorm weeks' mention networks, and the dissimilarity between all of these networks and the firestorm mention networks, still does not show whether or not a firestorm had an effect on the network structure. To investigate this, we constructed comparison 'panels' by randomly sampling from the decahose users who tweeted during the week of the event but not about the firestorm itself. We again generated mention networks across five weeks for these users. This time, we looked at the Jaccard indexes between weeks -1 and +1, and between weeks -2 and +2, and compared the distribution of these Jaccard indexes from networks of randomly sampled users to networks of firestorm participants. We found that the difference in means was not significant. That is, the way in which firestorms may change the mention networks of participants is not significantly different from the churn in networks that we would expect by random chance.

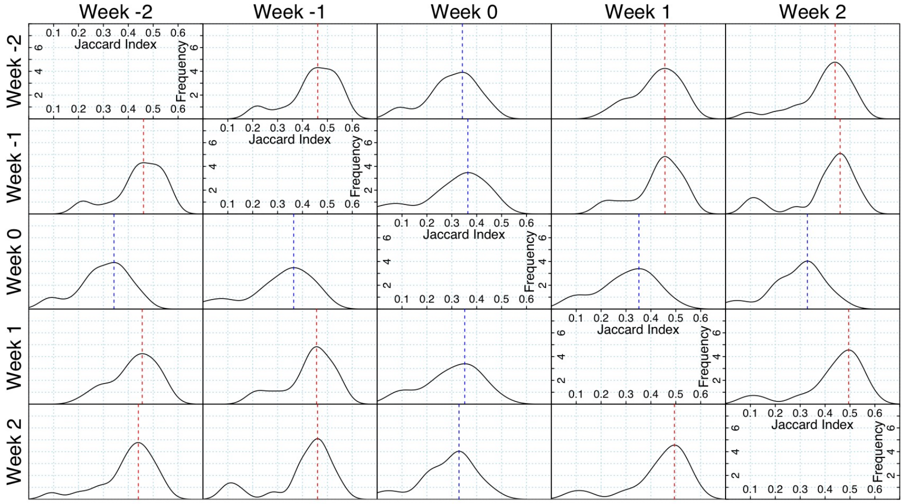


Figure 3.8: Distributions of the Jaccard index of edges between the mention networks two weeks before, one week before, during, one week after, and two weeks after the firestorms. Vertical lines are put at the mode of each distribution. The matrix is symmetric; this redundancy is provided for ease in vertical comparisons. The figure shows that the networks after the firestorms resemble much more the networks before the firestorm than during the firestorm.

3.4 Discussion

Our research question was about the relationship between social ties and firestorm participation. We find first that the mention networks pre- and post-firestorm are more similar to each other than to the mention network of the week of the firestorm. If firestorms emerged from existing networks, we would expect to find more similarity between firestorm mention graphs and pre-firestorm mention graphs. Conversely, if firestorms had created lasting links among participants, we would expect to find more similarity between firestorm mention graphs and post-firestorm mention graphs. Instead, we found low similarity between firestorms and other weeks. We further find by comparison to a randomly sampled group that we cannot find the firestorm had *any* discernible impact on patterns of discussion. Going back to our theoretical motivations, it seems that at least among the firestorms we sample, we see no evidence of the type of social change associated with action that has biographical consequences on participants. This suggests that, at least along this dimension, firestorms should not be a source of anxiety for targets nor a source of satisfaction for opponents; firestorms in general do not create the conditions to lead to larger and more long-term actions, at least among the mass of participants.

3.5 Conclusion

We have identified that across events identified as 'firestorms,' there is a departure from otherwise regular patterns of social interactions. Since both pre- and post-firestorm mention networks are different from firestorm mention networks, but the pre- and post-firestorm networks are similar to each other, it seems that the firestorms do not have a significant impact on communities. From our theoretical background, this finding suggests that firestorms will generally have little long-term impact. We believe that there are still interesting future research directions, including for basic research, including:

- Distinguishing sarcasm using ties and temporal clues;
- Firestorms as subsets of the Twitter ecosystem with different spam dynamics;
- Event detection and decay modeling of a specific, emotional and social type of event;
- Feedback effects on firestorms of simultaneous media coverage;
- Identifying the target of negative statements; for example, negative #CancelColbert tweets may be angry with Colbert or with the campaign.

Momin M. Malik, Hemank Lamba, Constantine Nakos, Juergen Pfeffer. ”Population Bias in Geotagged Tweets” Ninth International Conference on Web and Social Media (ICWSM) 2015.

CHAPTER 4

UNDERSTANDING BIAS IN GEOCODED DATA

Geotagged tweets are an exciting and increasingly popular data source, but like all social media data, they potentially have biases in who are represented. Motivated by this, we investigate the question, ‘are users of geotagged tweets randomly distributed over the US population’? We link approximately 144 million geotagged tweets within the US, representing 2.6m unique users, to high-resolution Census population data and carry out a statistical test by which we answer this question strongly in the negative. We utilize spatial models and integrate further Census data to investigate the factors associated with this nonrandom distribution. We find that, controlling for other factors, population has no effect on the number of geotag users, and instead it is predicted by a number of factors including higher median income, being in an urban area, being further east or on a coast, having more young people, and having high Asian, Black or Hispanic/Latino populations.

‘Geotagged’ or ‘geocoded’ tweets, where users elect to automatically include their exact latitude/longitude geocoordinates in tweet metadata, provide data that are:

- High-quality: geotagging is automated, so there are fewer chances of data error such as from user specification [91, 118];
- Precise: geotags are down to a ten thousandth of a degree in latitude and longitude;
- Richly contextual: geotags are connected to tweets with all their temporal, semantic, and social content;
- Easily available, through the Streaming API;
- Large: using the Streaming API, a researcher can build a collection of tens of millions of tweets.

Unsurprisingly, this makes them an enormously attractive source for studying a wide range of human phenomena [123]. Existing works have used geotagged tweets to study

- mobility patterns [48, 310],
- urban life [62, 76],

- transportation [291],
- natural disasters, crises, and disaster response [151, 176, 205, 257, 272], and
- public health [82, 213, 272]

as well as the interplay between geography and

- language [66, 123, 143],
- discourse [171],
- information diffusion and flows [136, 285],
- emotion [201], and
- social ties [48, 264, 274].

Furthermore, maps of geotagged tweets tend to look remarkably similar to maps of population density (figs.4.1 and 4.2; see also [171], even if there are differences at a finer scale (figs. 4.3a and 4.3b). This naturally leads to the question: are Twitter users who send geotagged tweets (henceforth, 'geotag users') randomly distributed over the population? This is a critical question because, if users who elect to geotag are systematically different from people in general, the results of studying geotagged tweets will not have external validity.

We used the Twitter API to get a collection of 144,877,685 geotagged tweets from the contiguous US, from which we extract 2,612,876 unique twitter handles. We uniquely assign each handle to a *block group*, a geographic designation of the US Census Bureau that is the smallest geographic unit for which Census data is publicly available. We then link the counts of unique geotag users per block group to the 2010 Decennial Census population counts per block group, and create a statistical test for the null hypothesis that geotag users are randomly distributed over the US population. We find sufficient evidence to reject this null. Using other Census data, we then use a Simultaneous Autoregressive (SAR) model to test some candidate explanatory factors and investigate what is nonrandom about this distribution. This is, to our knowledge, the first paper to use statistical testing to establish population bias along multiple dimensions in geotagged tweets across the entire United States.

4.1 Background and Related Work

Our study relates to an increasing body of work about biases in who and what is represented in social media data. The first work with Twitter data was by [200], who found an overrepresentation of populous counties and an underrepresentation specifically of the Midwest, an undersampling in counties in southwest with large Hispanic populations, an undersampling in counties in the south and midwest with large Black populations, and an oversampling of counties associated with major cities with large White populations. However, these findings come from interpretations of distributions and county-level cartograms, rather than from statistical testing, and they rely on the user-defined 'location' field, which has been shown to have many inconsistencies [91, 118]. Our study is on the one hand deeper because we use the far higher resolution of block groups and carry out statistical tests, but on the other hand not as general because our findings apply only to characteristics of *geotag users* within the US population rather than to geotag users within the Twitter population, or to Twitter users within the US population. Also worth noting

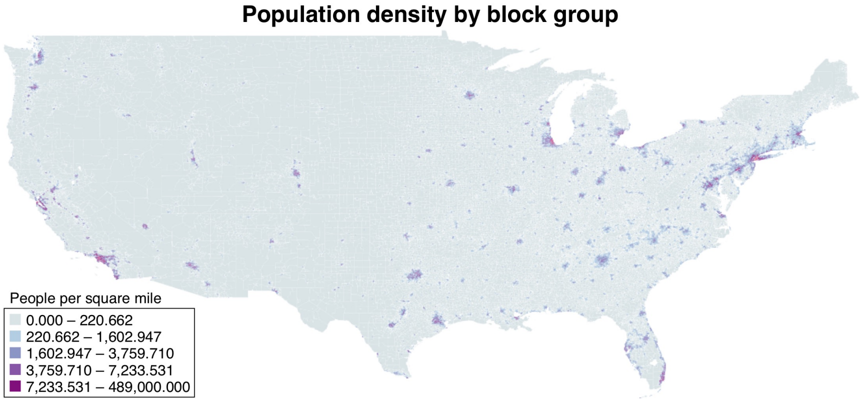


Figure 4.1: Quintiles of population per square mile by ‘block group’ (see below) in the 2010 Decennial Census.

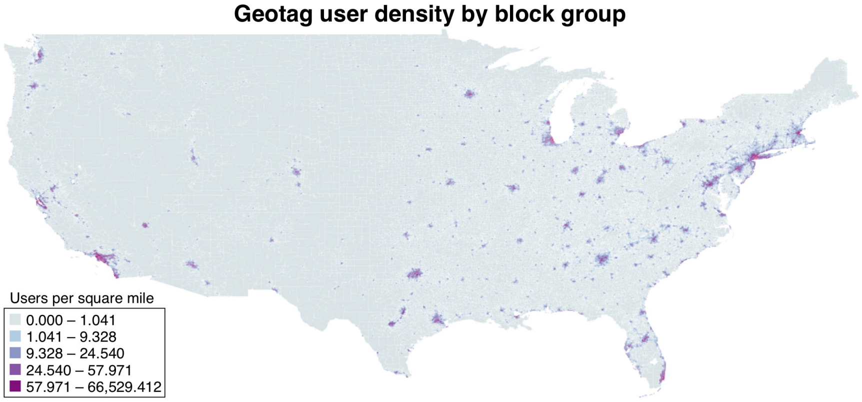
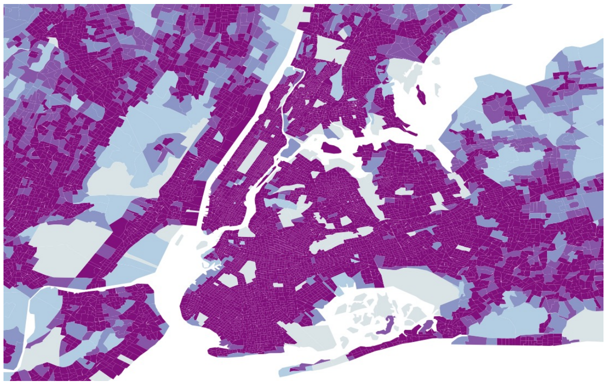
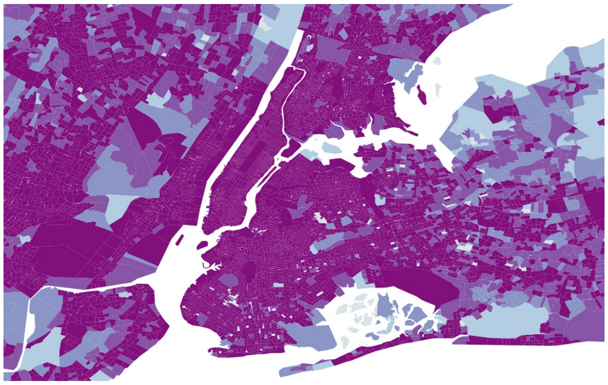


Figure 4.2: Quintiles of geotag users, uniquely assigned (see ‘mobile users’ below) per block group, divided by block group area.



(a) Detail of fig.(4.1) for New York.



(b) Detail of fig.(4.2) for New York.

is that Twitter has undergone large changes since the data used by [200], both in the governance and management of the platform itself [284] and in patterns of user behavior [181].

More recently, [117] investigated urban biases across the US. Collecting 56.7m tweets from 1.6m users over a 25-day period in August and September 2013 and comparing it to Census data, they use a method of calculating a reduced effective sample size in order to correct for spatial dependencies. From this they calculate ratios of users per capita and find a bias towards urban areas, with 5.3 times more geotagged tweets per capita in urban regions as in rural ones, a magnitude even more pronounced in Foursquare data. [183] investigate biases across a number of factors, focusing on the Greater London area. Using work on forename-surname pairs identifying gender, age and ethnicity, they parse usernames and other profile information to get a collection of estimated names, which they then compare to the 2011 UK Census and find an overrepresentation of young males, an underrepresentation of middle-aged and older females, an overrepresentation of White British users, and underrepresentation of South Asian, West Indian, and Chinese users, although tests of significance are not applied.

Coming from another methodological direction, a nationally representative survey study of smartphone owners (n=1,178) by Pew [314] looks at the demographics of location service users. Overall, 12% of those surveyed reported using what Pew terms 'geosocial' services (which includes geotagged tweets, and excludes informational services like Google Maps). Interestingly, the survey finds the the most frequent users of geosocial services are those of low*est* income and middle income; those of low*er* income use it less, and those of upper income use it least. More 18-26 year olds use geosocial services than older users, and almost double the proportion of hispanic (English- and Spanish-speaking) smartphone owners user geosocial services as compared to white and black (both non-hispanic) smartphone owners. However, out of the respondents who specified which geosocial services they use (n=141), most reported using Facebook (39%), Foursquare (18%) or Google Plus (14%); only 1%, or 1 respondent, used Twitter's geosocial services (i.e., geotagged tweets), such that it is not possible to make inferences about geotag users from the results of this study.

Our paper is answering the general call for stronger methodological investigations about the nature of population representation in social media data [245, 283], as well as the specific call for combining geographic data from user-generated sources with non-user-generated sources, such as Twitter data with the Census [52].

4.2 Data Collection

4.2.1 Geo-Coded Twitter Data

From Twitter's Streaming API, we collected 144,877,685 tweets from April 1 to July 1, 2013 using the geographic boundary box $[124.7625, 66.9326]W \times [24.5210, 49.3845]N$. This covers the contiguous US (i.e., the 48 adjoining US states and Washington DC but not Alaska, Hawaii, or offshore US territories and possessions). Consequently, all our tweets are geo-coded with lat/long GPS coordinates. As [204] report from the Twitter Firehose, about 1.4% of tweets are geotagged; and elsewhere [206] they report the Streaming API is more likely to be biased when the response to a query exceeds 1% of the total volume of tweets. Given also that North America accounted for only 22.32% of geotagged tweets in their collection, a fraction consistent with what [181] report finding in a collection of decahose data covering the time period we consider, it is reasonable to assume that the use of the Twitter API to collect tweets geotagged in the US covers all or nearly all of geotagged tweets within the given time frame and geographic bounds.

Since the distribution of geotagged tweets over geotag users is characteristically long-tailed (fig. 4.4), with a minority of users sending out the majority of tweets, we decided that the relevant quantity was the number of geotag users rather than the number of tweets. We identified 2,612,876 unique user accounts in our data, which is the basis of our analysis.

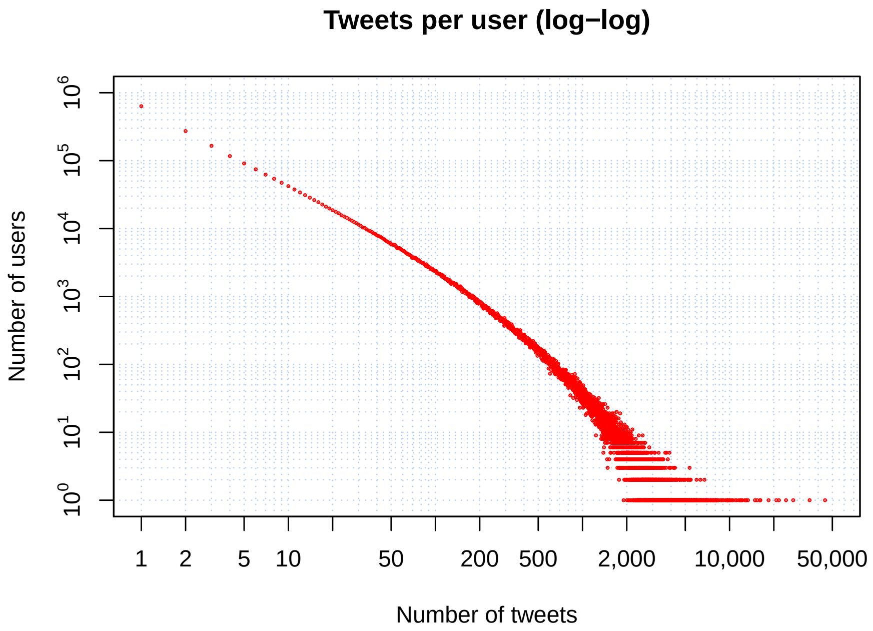


Figure 4.4: The usual long-tailed distribution of the number of users who have tweeted a certain number of tweets. Because of this skew, we focus on unique users alone, and ignore the volume of tweets.

4.2.2 Geospatial Data

The contiguous US plus Washington DC include 215,798[1] block groups (2010 specification) which range in size from .002 square miles to 7503.21 square miles. Block groups are designed by the Census Bureau to have roughly comparable population sizes. We verified this by noting that, in log scale, the distribution of populations per block group has a symmetric distribution and stable variance. Each block group has a unique identifier, the 12 digit *FIPS Code*, consisting of identifiers for state (first two digits), county (next three digits), tract (next six digits), and block group (last digit). For every state, the US Census Bureau provides geographic boundary files ('shapefiles') that includes the GPS coordinates of the borders of every block group within the state. We combined the shapefiles of the 48 contiguous states and the District of Columbia, deleting 364 block groups representing bodies of water (identifiable by being coded as having

[1] Probably due to a rounding error in geographic calculations, we lost three small island block groups (2 in Florida, 1 in New York), such that our n = 215,795.

zero area, and having a FIPS code ending in zero[2]). With Python code (utilizing the `shapely` package) we identified the Census block group into which each tweet fell.

4.2.3 Socioeconomic Data

While the ideal would be to have rich and timely demographic data about the users who sent the tweets in our data, this is not realistic to collect for 2.6m users. But by aggregating data at the level of block groups, we can link Twitter data to the enormously rich demographic data the Census Bureau makes available at this level. We primarily use data from the 2010 Decennial Census, which we supplement with median income (not available in the Decennial Census) estimates from the 2009-2013 American Community Survey. For this ACS data, there were 1,224 block groups with missing values for median income, few enough that we filled these out as zeros rather than using imputation or smoothing. We also set 21 block groups with the value "2,500-" to 2,500, and 2,651 block groups with the value "250,000+" to 250,000. The 2009-2013 ACS had 54 block groups in the contiguous US whose boundaries (and FIPS) codes were from the 2000 Census, for which we found equivalent block groups in the 2010 Decennial Census to which to map. While the ACS 1-year estimates are more timely, they are more sparse and only at the county level [13], and we decided to prioritize the accuracy and completeness of values in the Decennial Census for this analysis. We similarly decided to not use the ACS 2009-2013 estimates for population quantities as there was more missing data, and there was high correlation between the 5-year estimates and 2010 Decennial Census figures across variables (generally around .95). Still, prioritizing timeliness over completeness, and looking at the county level with 2013 ACS 1-year estimates, may be the focus in future analysis.

4.2.4 Mobile users

Our construct of interest is the *number of potential geotag users*, for which population is the available proxy; there are cases where there are more geotag users than population, which points to tourists or, more generally, mobile users, as a complicating factor [117].

[117] provide a useful review of techniques to uniquely assign users to a single geographic region. They identify two candidate techniques: temporal, where a user must send at least two tweets a set number of days apart in a region for the user to be located uniquely in that region, and 'plurality rules,' where the most frequently tweeted-from region is taken as the unique location of the user. Checking the 'location' field fails because of the low quality of the information there [118]. As one other option, [294] use the location of the first geotagged tweet sent by a user as the location of the user. This is the simplest, but also has no motivation beyond convenience.

Despite the drawbacks of plurality not accounting for people local to two regions, our comparison is with the US Census which also does not account for this possibility. However, another problem is that foreign tourists are not counted in the US Census (unlike domestic tourists, who reside in some US block group), and of which there were 70m in the US in 2013[3]. This is substantial when compared to the total 2013 US population of 316m[4] (of which 307m are counted

[2] https://www.census.gov/geo/reference/gtc/gtc_bg.html

[3] http://travel.trade.gov/view/m-2013-I-001/table1.html

[4] http://data.worldbank.org/indicator/SP.POP.TOTL

in the block groups we use). If many foreign tourists send geotagged tweets, it would introduce unaddressed bias; since our data collection only had geotagged tweets in the US, short of massive additional data collection we are unable to identify foreign tourists (such as by looking at the proportion of geotagged tweets outside of the US). This is a potential problem in our analysis that may be a topic for clarification in future work.

Additionally, we filter users by the number of tweets, considering only those with a certain number of tweets.[5] As the distribution of tweets per user (fig. 4.4) is smooth and has no natural break point, we arbitrarily pick 5 and 10 as cutoffs to use alongside all users.

4.3 Statistical Models

4.3.1 Random distribution over population

The basic relationship in which we are interested is between population and geotag users. In order to make a concrete test for random distribution, we suggest a model where there is a linear relationship between the population count and the number of users, i.e. users are drawn from the population at a constant rate subject to some noise. We can imagine the noise is heteroskedastic, which suggests the following data-generating process over population P, users U, and mean-zero noise term ε:

$$U = \alpha P + \varepsilon P \tag{4.1}$$

We transform both users and population to stabilize their variances, so this then becomes

$$\log U = \log \alpha + \log P + \log \left(1 + \frac{\varepsilon}{\alpha}\right) \tag{4.2}$$

Then, consider the linear model

$$\log U = \beta_0 + \beta_1 \log P + \varepsilon' \tag{4.3}$$

If eqn. (4.1) described the true data-generating process, from eqn. (4.3) we should get that $\hat{\beta}_1 = 1$, and then $\exp(\hat{\beta}_0)$ would estimate the value of the proportion α. That is, the $\log \alpha$ term is the intercept of the regression of $\log P$ onto $\log U$, and $\log \left(1 + \frac{\varepsilon}{\alpha}\right)$ is a mean zero error term now independent of P, and we have a null hypothesis $H_0 : \beta_1 = 0$. While this may seem unrealistic as a null model, other quantities that we would believe are randomly distributed proportional to population indeed match this. For example, we regressed log population onto log males and found it to be meaningful (presented below under results). With this validation, we argue that the model of eqn. (4.1) is a reasonable way of representing a quantity being randomly distributed over the population. Note that our interest is not in fitting this specific model and interpreting the parameters, but just having a way to test the null hypothesis of random distribution. Note also that we originally sought to compare log population density to log geotag user density as a way of treating measures on different block groups as equivalent (given that block groups are already designed to somewhat control for the variance in population density), but found that it produced excellent fits that did not disappear when the data was shuffled, suggesting that the dividing by area created artifactual relationships.

[5] We thank an anonymous reviewer for this fruitful suggestion.

4.3.2 Model specification

For comparison with analyses of race and Hispanic populations [200, 314], we use Census variables[6] P0030001 through P0030008 and P0040001 through P0040003. For comparison with analyses by age [183, 314], we use P0120003 through P0120049 and aggregate across gender into the same age bins as in [314]. Existing analyses by sex [183, 200, 314] is based on name-based inference or survey data; we decided that, while the Census does have sex data, the even distribution of sex across the US means that the sex ratio of a block group is not a meaningful proxy for geotag users who live there. For comparison with analyses of urban and rural populations [117, 314], we use P0020002 through P0020005.[7]

Thus, in total, we include terms for populations, the black population, the Asian population, the Hispanic/Latino population, the rural population, and respective populations of people ages 10-17, 18-29, 30-49, 50-64, and 65+. For all of these, we stabilize variance with a log transformation with add-one smoothing. We include median income [314], and test for a northern/eastern effect by including the (demeaned) latitudes and longitudes of block group centroids, and for a coastal effect by including terms for latitude and longitude squared.

Spatial autocorrelation: Discretization into uneven geographic units (as block groups certainly are) can cause statistical artifacts. Specifically, if the divisions do not correspond to the contours of the underlying spatial process (and there is little reason to believe they would), there will be dependencies between proximate geographic areas, and not accounting for this can inflate the R^2 statistic, shrink standard errors, and give misleadingly significant results. We use the standard statistic for measuring spatial autocorrelation, Moran's I,

$$I = \frac{n}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(X_i - \overline{X})(X_j - \overline{X})}{\sum_i (X_i - \overline{X})^2} \tag{4.4}$$

This is the empirical covariance, appropriately normalized, of the values of variable X between geographic units i and j. $W = [w_{ij}]$ is an $n \times n$ matrix of weights, discussed below. Rather than exploring autocorrelation in individual variables, we look for spatial autocorrelation in the residuals of a linear model [15]. For management of spatial data and implementation of computation and estimation for spatial models, we used the R package `spdep` [24, 25].

Weights Matrix: Measuring spatial autocorrelation requires a 'weights matrix' of adjacencies between geographic units. There are multiple ways to generate this, and the choice of how to do so represents a substantive decision based on the problem at hand [78]. However, given that we do not know in advance the form of the spatial autocorrelation, in practice we can test for autocorrelation over different choices of weights matrices to see which is most appropriate [16]. Thus, we consider the following weights matrices:

- Queen contiguity (regions sharing a corner or edge are adjacent, equivalent to 8-connectivity in image processing);
- Rook contiguity (regions sharing an edge are adjacent, equivalent to 4-connectivity in image processing)

[6] http://api.census.gov/data/2010/sf1/variables.html

[7] The Census API returned zero values for these, so we manually downloaded the variables of "P2. URBAN AND RURAL" for each state individually from factfinder.census.gov.

- k-nearest-neighbors for $k = \{2, 3, 4, 5, 6, 7, 8\}$, calculated from the midpoints of block groups.

For the contiguity cases, we consider both row-normalized (which normalizes the ‘effect’ of each neighboring unit such that they sum to one) and binary (which gives greater possibility for autocorrelation between a unit and its neighbors for units with more neighbors).

4.3.3 Spatial errors model

We model the relationship between population and geotag users using a Simultaneous Autoregressive (SAR) model, which is where one or more terms in the regression are correlated with itself. The main autoregressive model assumes that the residuals of unit i are correlated with the residuals of those units j adjacent to i, which is known in econometrics literature as a spatial errors model. The adjacencies are indexed exactly by the terms of the weights matrix. This gives the following two equations,

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{u} \tag{4.5}$$

$$\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \varepsilon \tag{4.6}$$

where u are the correlated residuals, $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ are the uncorrelated error terms, and the coefficient λ is the ‘spatial multiplier’ that captures the strength of the spatial autocorrelation [14]. While there are other SAR models, we use spatial errors as the simplest to interpret and the most appropriate for our purpose.

4.4 Results and Discussion

4.4.1 Observational Results

The block groups with the highest number of distinct users (before users are assigned uniquely) are major international airports and major tourist attractions (table 4.1).[8] The inclusion of several international airports on the list suggests that geotagging tweets during the process of travel is a common user behavior. There were some areas with zero population but nonzero users; out of these, the ones with the highest counts of distinct users are mostly the same: major airports and parks.[9]

Conversely, there were only 67 block groups from which nobody sent geotagged tweets; only 30 of these also had no population (these were national forests, minor airports, areas off highways, etc.). Of those that did have a population, the most populous was a block group with a population of 4,854 within San Quentin State Prison in California. The second-most populous block group is also a Corrections Department building in Texas, and third is a state prison in California (although not all prisons lack geotag tweet users; the block group of Rikers Island in New York has geotagged tweets from 22 users).

[8] Block groups may be looked up by their FIPS code at http://www.policymap.com/maps

[9] Interestingly, Central Park has a nonzero population (of 25), as do some airports. Some other tourist attractions (e.g., Universal Studios) also appear.

Table 4.1: Block groups from which the most users have sent geotagged tweets.

FIPS code	Users	Description
32 003 006700 1	28,280	Las Vegas Strip
06 037 980028 1	23,100	Los Angeles Int'l Airport
32 003 006800 4	16,748	McCarran Int'l Airport
13 063 980000 1	15,481	Atlanta Int'l Airport
12 095 017103 2	15,392	Walt Disney World
36 081 071600 1	15,067	JFK Int'l Airport
11 001 006202 1	14,906	National Mall
36 061 014300 1	14,605	Central Park
06 059 980000 1	14,576	Disneyland
17 031 980000 1	13,610	Chicago Int'l Airport

Out of the 2,612,876 unique users we identified, 2,216,219 (84.82%) had a single block group from which they tweeted most frequently. The others had ties for which block group was the highest; for these users, we uniquely assigned them to one of their block groups by randomization. We tried analyses on just the 84.82% as well, but found it made little substantive difference in the results.

In the terminology of [104], the most active accounts belong to 'non-personal users.'[10] In this case, the most active tweeter (44,624 tweets) seems to be a commercial service for travel, the second-most active (35,025) is an automatic news updater in Florida, etc. Starting from the 13th most active tweeter, with 12,922 tweets, there were accounts that appeared on inspection to be personal ones. As for number of block groups traversed, the top 'traveler' (23,547 block groups) is the same as the top tweeter, and others are similarly non-personal users. Across block groups, it is not until the 18th most mobile user, traversing 1,209 block groups, that there is a personal user.

How much mobility is there between units? Figures 4.5 and 4.6 show respectively that while there is minimal mobility between states, with only 22.39% of users sending geotagged tweets from more than one state and only 7.83% send from more than 2. However, there is a great deal of mobility between (possibly neighboring) block groups, with 65.24% of users sending geotagged tweets from more than one block group.

How well does unique assignment do? As one check, we consider the ratio of geotag users to population; there are 509 block groups where this ratio is greater than 1 (for users with 5 or more tweets only, there are 353, and for users with 10 or more tweets only, there are 290), indicating either the failure of population as proxy for potential geotag users or of the method of assigning mobile users. As we found the block groups with the largest ratios to be airports, it seems to be a case of the latter.

[10] They find that only 2.6% of geotag users are non-personal. This should be small enough to have no effect on results, so we did not employ filtering. However, this may be considered in a future work.

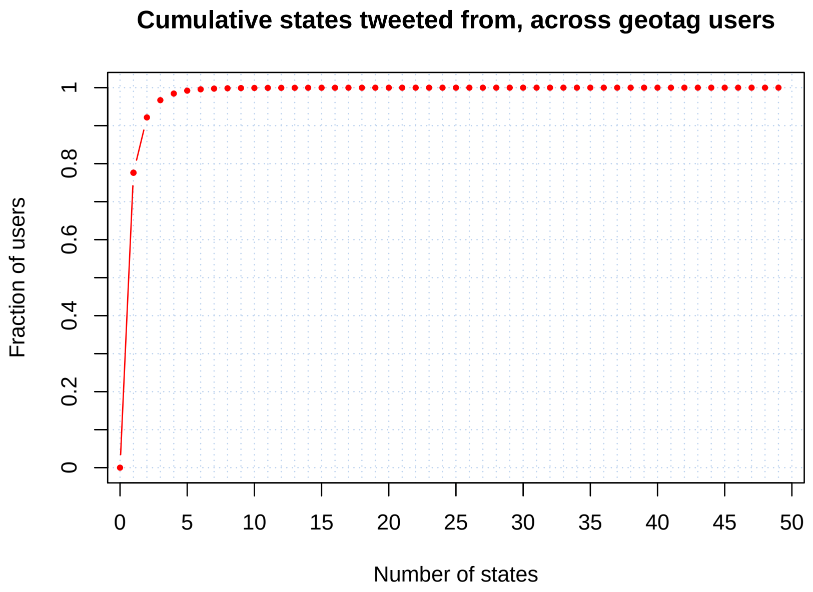


Figure 4.5: A full 77.61% of geotag users in our set tweeted only from one state, and having tweeted from 5 or fewer states accounts for 99.21% of users.

4.4.2 Bivariate regression model

We first test our null hypothesis of a linear regression yielding a coefficient of 1 to the logarithm of the population. Looking at the plot of the relationship of the logarithm of the two (fig. 4.7), there is a faint linear relationship, although the slope does not appear to be 1. An OLS regression fits slope $\hat{\beta}_1$ = .4916 (.002996) and intercept $\hat{\beta}_0$ = -1.219 (.02143),[11] although we should recall that the standard errors are not reliable under spatial autocorrelation.

Compare this plot to the plot of our test case mentioned earlier, the distribution of males over the population, pictured in fig. 4.8. The true ratio of males to total population across the block groups we consider is .4915; according to our model, the exponential of the intercept should be this, and the coefficient of the log population term should be 1. Indeed, log(.4915) is within the 95% confidence interval (log(.4914), log(.4962)), and 1 is just outside the 95% confidence interval (.9980, .9994), but this is without accounting for how spatial autocorrelation shrinks estimated standard errors. The R^2 value of this model is also impressive at .975, although under spatial autocorrelation R^2 is inflated thereby not interpretable. Overall, our model fits the relationship of males to population exactly as we would expect it to fit to something randomly distributed over the population.

Using this as a validation of our statistical test, we can strongly reject the null hypothesis that $\hat{\beta}_1 = 1$ even without correcting for spatial autocorrelation. And the R^2 value for this regression is a paltry .109, too small to worry about being inflated. Thus, we can conclude that geotag users are not randomly distributed over the US population, and indeed that the population count is not

[11] Filtering for only those users who have 5 or more tweets and for those users with 10 or more tweets, the respective fitted slopes are .5192 (.002932) and .5136 (.2786).

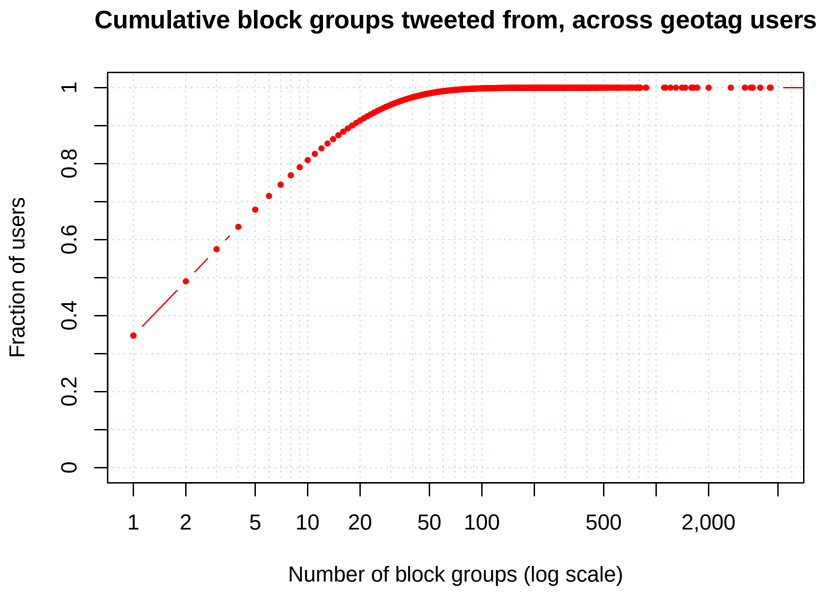


Figure 4.6: 34.76% of geotag users tweeted only from one block group. 27 or fewer block groups were 95%, 50 or fewer block groups were 99%. One outlier at 23,547 excluded.

very informative about the number of geotag users.

Weights matrix and spatial autocorrelation.

Testing the residuals in our basic model for spatial autocorrelation using Moran's I against all weights matrices considered above, we find the results reported in table 4.2.

We found identical results of Moran's I for binary weights matrices and row-normalized weights matrices in the k-nearest neighbor case. For the two contiguity cases, row normalization made a difference, and we list both values. In all cases, an asymptotic test against the expected value of 0 was significant at $p < .0001$. The autocorrelation in the population-user model is stronger than in the 'null' population-male model. It appears, then, that the spatial autocorrelation is strong enough that the choice of weights matrix is not critical. For the population to user model fit on counts of users with 5 or more tweets, or 10 or more tweets, the spatial autocorrelation was similar (generally lower, but still higher than the autocorrelation of population vs. male).

4.4.3 Spatial errors model

The maximum likelihood method of fitting a SAR model involves computing the log determinant of the $n \times n$ matrix $|I - \lambda W|$, which is infeasible at our n of over 200,000. An alternative method finds the log determinant of a Cholesky decomposition of $(I - \lambda W)$, although this then requires W to be a symmetric matrix [26]. Since all of the candidate weights matrices picked up spatial autocorrelation at a significant level, we use a binary contiguity weights matrix. We tried both

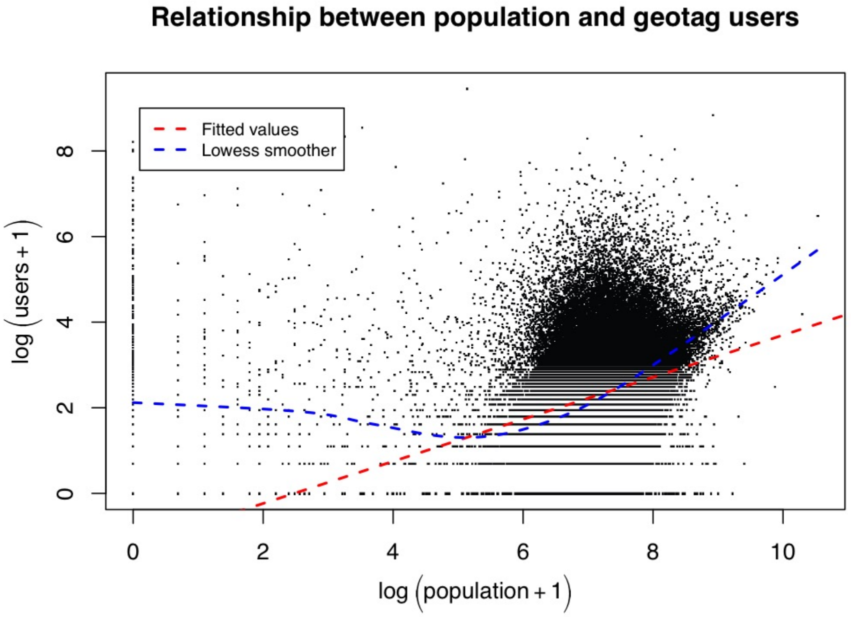


Figure 4.7: Eliminating zero-count observations reduces the artifacts visible at $x = 0$ and $y = 0$ but does not substantially change the fit.

Rook and Queen, and they gave comparable fits, so we report only for Rook (table 4.3).

The spatial multiplier term is significant, although neither the coefficients nor the standard errors are substantively different than the previous model. However, calculating Moran's I on the residuals of this model gives a value of -.02367, with a p-value of 1, meaning we have successfully controlled for spatial autocorrelation.

We then investigate the full model specified above. We interpret this model in the standard way: for a log transformed explanatory variables X_i, a 1 percent change will predict a β_i percent change in $\mathbf{Y}$. We present the results of the regression on counts of only those users with 5 or more tweets. This is shown in table 4.4.

As before, testing for spatial autocorrelation finds no significant amount, with a p-value of 1. Here we see that, after controlling for other factors, population loses its significance (this also points to the benefits of using a SAR model, as under OLS the population term is significant). The term for area included as a control is significant, with a one percent rise in block group area predicting a 15.56% rise in geotag users. It seems here that size overcomes the effects of population density (as mentioned above, block group population has stable variance only in log scale even though block groups are designed to enclose populations of roughly comparable size). Consistent with survey findings [314], a 1% larger Hispanic/Latino population predicts 1.533% more geotag users. However, the effect size is smaller than either that of the Asian population (a 1% rise predicting an 11.12% rise in geotag users) and, in contrast to survey findings, that of the Black population (a 1% rise predicting a 4.29% rise in geotag users). This might point to the Pew sample not including enough Twitter users, as there is an active Black community on Twitter that is gaining scholarly attention [51, 74, 256]. The latitude, both in linear and quadratic effects, is not significant; however, the longitude is significant, pointing first to block groups

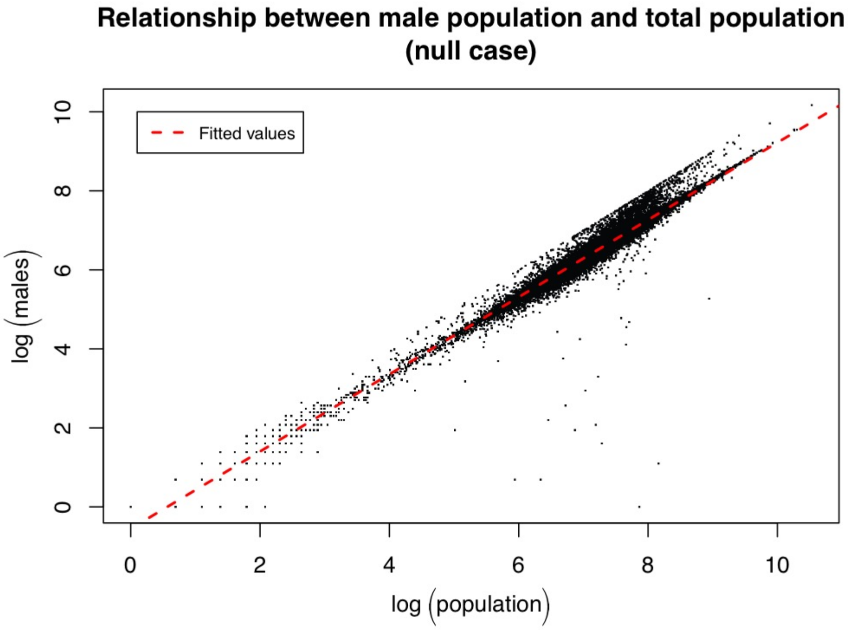


Figure 4.8: The relationship between males and total population behaves exactly as we expected of a quantity randomly distributed over the population, making it an effective null model against which to compare the observed distribution of geotag users.

further east having more geotag users, and second (from the positive sign of longitude squared) to a coastal effect where block groups on both the east and west coasts have more geotag users than in the center of the US. While we tried to test for nonlinearity in income, inclusion of a squared term for median income made the matrix computationally singular; however, inspecting the bivariate relationship did not yield any evidence for a nonlinear effect, and the linear effect is weak (a $10,000 rise in the median income predicts a 1.66% rise in the number of geotag users). Consistent with findings about urban biases [117], we find that a 1% higher rural population predicts a 5.72% decrease in the number of geotag users. Lastly, also consistent with survey findings, 18-29 year olds are the most active geotag users, with a 1% higher population of this age group predicting 39.16% more geotag users. There is also a strong negative effect for the population of ages 50-64, with a one percent change predicting 17.93% fewer geotag users, but the teenage population surprisingly predicts fewer geotag users. Also surprisingly, there was a significant and positive effect from the population people 65 and older. These might be due to more complex interactions such as mixed populations. As is usual with logarithmic dependent variables, the intercept is not particularly interpretable as it would be a prediction for a block group at the center of the US with a population of 1.

Running the SAR model using all users, instead of just those with 5 or more tweets, produces similar results, except that log population is significant with coefficient -.04196 (.007858); this suggests a nonlinear effect, and indeed, an added squared term for the log population came out as significant and positive at .06329 (.0008394). This points to some noise for those people who only ‘try out’ geotagged tweets but do not adopt their use that disappears if we maintain a minimum tweet threshold. When running the model on only those users with 10 or more

Table 4.2: Selected Values of Moran's I in residuals

	Population vs Users	Population vs Male
2nn	.3699	.2336
4nn	.3550	.2142
6nn	.3398	.1996
8nn	.3270	.1883
Rook	.4166 (b)	.2125 (b)
	.3992 (rn)	.2201 (rn)
Queen	.4151 (b)	.2097 (b)
	.3919 (rn)	.2154 (rn)

For the Rook contiguity case and the Queen contiguity case, binary (b) and row-normalized (rw) weights gave different values.

tweets, results are again similar except the longitude squared term is no longer significant (p = 0.1870), and the latitude term becomes significant (p = 0.02017). This might be from the coasts having more users who try out geotagged tweets for a longer period of time before choosing not to continue. These subtle differences point to opportunities for modeling the demographics of different types of users (as determined by number of geotagged tweets or other factors), although we do not explore them more here.

4.5 Conclusion

Geotag users are not representative of the US population. Despite the volume of geotagged tweets and their impressive coverage (there were only 67 block groups out of 215,795 with no geotagged tweets), the users who send geotagged tweets are nonrandomly distributed over the population in subtle ways. These include predicable and already established biases towards younger users, users of higher income, and users in urbanized areas, as well as surprising biases towards Hispanic/Latino users and Black users that, in the latter case, have not seen in large-scale survey research. We also demonstrate an unsurprising but previously unreported coastal effect, where being located on the east or west coast of the US predicts more geotag users. Geotag users may not be a random sample of the population of any given block group, but given the fine level of detail and large-scale demographic variability, the demographics of a block group is a reasonable proxy for the demographics of geotag users located in that block group. Certainly, even with complications of uniquely assigning mobile users, it is enough to establish the nonrandom distribution of geotag users, and some candidate biases.

While from this study, we are unable to say whether or not geotag users are representative of the *Twitter* population, the more interesting question we address is whether geotagged tweets can be a useful proxy for the *general* population within the US. This is a critical question because geotagged Tweets are an enormously popular source of data for studying a wide variety of social and human phenomena. For future work, we emphasize that findings using geotagged tweets

Table 4.3: Spatial errors basic model, binary Rook contiguity

	Dependent variable:
	log(user + 1)
log(population + 1)	.4401*** (.002655)
Intercept	-1.138^{***} (.01890)
$\hat{\lambda}$:	.1107***
LR test value:	73,375
Numerical Hessian $\widehat{\mathsf{se}}(\hat{\lambda})$:	8.4241e$-$06
Log likelihood:	$-222{,}020.8$
ML residual variance (σ^2):	.4206
Observations:	215,795
Parameters:	4
AIC:	444,050
Note:	***p<.0001

should not be assumed to generalize, and conclusions should be restricted only to geotag users with their population biases.

4.6 Future Directions

There are a number of directions for future work. One is to connect tweets to lower-resolution and lower-accuracy but more current 2013 ACS 1-year county-level estimates. Others are to see the effect of filtering out non-personal users, and to build ways to filter out foreign tourists and better uniquely place geotag users in the block group that is likely to be their residence. Modeling demographic differences between users of different levels of use is also possible with this data. We have applied one spatial model, but spatial modeling is a rich area with many other available techniques. For example, there are also relevant disease mapping models that break down incidence by various demographic strata [26] that would be appropriate here, as well as nonparametric models that might better capture irregular effects. Furthermore, we elected to not consider the temporal aspect; there is work on spatio-temporal modeling [136, 183, 213, 272] but it tends to be in the short-term window of a day or week. With reliable spatio-temporal models of how the prevalence of geotagged tweets per block group changes over longer periods of time and a better understanding of the demographic characteristics towards which geotag users are biased, we may be able to create models to provide a rapid and high-resolution proxy for demographic changes such as processes of gentrification, or urbanization, or urban decay; that is, utilize the very biases of social media data to make inferences about larger phenomena.

Table 4.4: Spatial errors full model, binary Rook contiguity, users with >5 tweets only.

	Dependent variable:	
	log(user + 1)	s.e.
log(population + 1)	-.01218	(.008081)
log(area)	.1556***	(.001760)
log(asian + 1)	.1112***	(.001576)
log(black + 1)	.04292***	(.001576)
log(hispanic + 1)	.01533***	(.002066)
latitude (demeaned)	-.006992	(.0007052)
longitude (demeaned)	.02306***	(.0002739)
latitude^2	-.0001641	(.00009505)
longitude^2	.00008777***	(.00001411)
median income ($10K)	.01661***	(0006857)
log(rural + 1)	-.05722***	(.001096)
log(ages 10-17 + 1)	-.09831***	(.003712)
log(ages 18-29 + 1)	.3916***	(.004423)
log(ages 30-49 + 1)	.06362***	(.006731)
log(ages 50-64 + 1)	-.1793***	(.006953)
log(ages 65 and up + 1)	.09675***	(.003940)
Intercept	1.3382***	(.1916)
$\hat{\lambda}$:	.1009***	
LR test value:	36,577	
Num. Hessian $\widehat{\mathsf{se}}(\hat{\lambda})$:	0.0003456	
Log likelihood:	−207,923.5	
ML resid. var. (σ^2):	.3755	
Observations:	215,795	
Parameters:	19	
AIC:	415,890	
Note:	***p<.0001	

Part III

Individual User Modeling

Hemank Lamba et al. ”From Camera to Deathbed: Understanding dangerous selfies on social media.” Eleventh International Conference on Web and Social Media (ICWSM) 2017.

CHAPTER 5

DETECTING DANGEROUS SELFIES BEHAVIOR ON SOCIAL MEDIA

Over the past couple of years, clicking and posting selfies has become a popular trend. However, since March 2014, 127 people have died and many have been injured while trying to click a selfie. Researchers have studied selfies for understanding the psychology of the authors, and understanding its effect on social media platforms. In this work, we perform a comprehensive analysis of the selfie-related casualties and infer various reasons behind these deaths. We use inferences from incidents and from our understanding of the features, we create a system to make people more aware of the dangerous situations in which these selfies are taken. We use a combination of text-based, image-based and location-based features to classify a particular selfie as dangerous or not. Our method ran on 3,155 annotated selfies collected on Twitter gave 82% accuracy. Individually the image-based features were the most informative for the prediction task. The combination of image-based and location-based features resulted in the best accuracy. We have made our code and dataset available at http://labs.precog.iiitd.edu.in/killfie.

With the rise in the amount and type of content being posted on social media, various trends have emerged. In the past, social media trends like memes [81, 173, 262], social media advertising [198], firestorm [160], crisis event reporting [246, 247], and much more have been extensively analyzed. Another trend that has emerged over social media in the past few years is of clicking and uploading selfies. According to Oxford dictionary, a selfie is defined as *a photograph that one has taken of oneself, typically one taken with a smart phone or web cam and shared via social media* [5]. A selfie can not only be seen as a photographic object that initiates the transmission of the human feeling in the form of a relationship between the photographer and the camera, but also as a gesture that can be sent via social media to a broader population [250]. Google estimated that a staggering 24 billion selfies were uploaded to Google Photos in 2015 [2]. The selfie trend is popular with millennials (ages 18 to 33). Pew research center found that around 55% of millennials have posted a ”selfie” on a social media service [4]. The popularity of selfie trend is so massive that ”selfie” was declared as the word of the year in

Figure 5.1: Left: Selfie took by a group of individuals shortly before they drowned in the lake. Right: Photograph of a girl taking a selfie on train tracks immediately before a train hit her.

2013 by Oxford Dictionary [3]. The virality of the selfie culture has also been known to cause service interruptions on popular social media platforms. For instance, the selfie taken by Ellen Degeneres, a popular television host, at the Academy Awards brought down Twitter website due to its immense popularity [1].

Selfies have proved instrumental in revolutionary movements [34], and have also known to help election candidates increase their popularity [18]. Many researchers have studied selfies for understanding psychological attributes of the selfie authors [156, 234], investigating the effect of selfies on social protests [34], understanding the effect of posting selfies on its authors [250], dangerous incidents and deaths related to selfies [23, 126, 269] and using computer vision methods to interpret whether a given image is a selfie or not [41].

Clicking selfies has become a symbol of self-expression and often people portray their adventurous side by uploading crazy selfies [100]. This has proved to be dangerous [23, 126, 269]. Keeping in mind the hazardous implications of taking selfies at dangerous locations, Russian authorities came up with public posters, indicating the dangers of taking selfies [99]. Similarly, Mumbai police recently classified 16 zones across Mumbai as no-selfie zones [300]. Through the process of data collection, we found 127 people have been killed since 2014 till September 2016 while attempting to take selfies. From 15 casualties in 2014 and 39 in 2015, the death toll due to selfies has reached 73 till September 2016. It has been reported that the number of selfie deaths in 2015 was more than the number of deaths due to shark attacks [6]. Some of the selfies that led to casualties are shown in the Figure 5.1. Given the influence of selfies and the significant rise in the number of deaths and injuries reported when users are taking selfies, it is important to study these incidents in detail and move towards developing a technology which can help reduce the number of selfie casualties.

In this work, we characterize the demographics and analyze reasons behind selfie deaths; based on the obtained insights, we propose features which can differentiate potentially dangerous selfie images from the non-dangerous ones. Our methodology is briefly explained in Figure 5.2. Specifically, the major contributions of the paper are as follows:

- **Data Characterization:** We do a thorough analysis of the selfie casualties, and provide insights about all the previous fatal selfie-related incidents.
- **Feature Identification**: We propose features that are easily extractable from the social

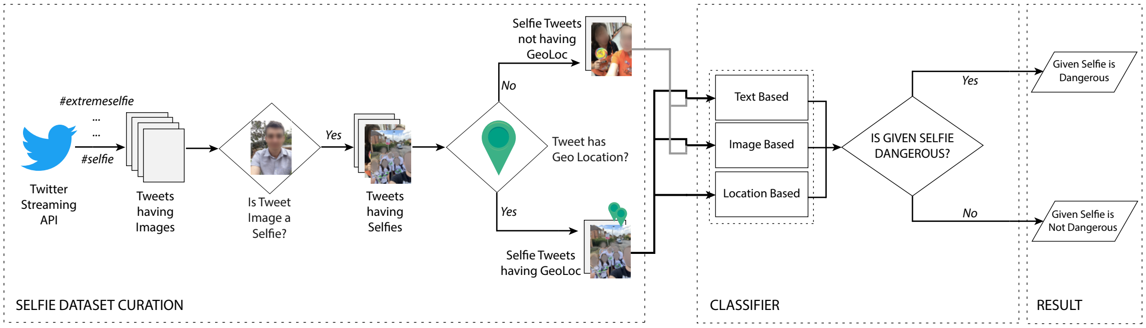


Figure 5.2: A brief overview of our approach - Tweets tagged with a geolocation are analyzed using text, location and image-based features. Whereas tweets without a geolocation are analyzed only using text and image-based features.

media data and learn signals which determine if a particular selfie is dangerous.

- **Discriminative Model**: We present a model that based on the proposed features can differentiate between dangerous selfies and non-dangerous selfies.
- **Real World Data**: We test our given approach on a real-world dataset collected from a popular social media website. We also test the efficacy of our approach in absence of certain features, a situation which is possible while working on such real datasets.

Furthermore, we believe our contributions could lead to generation of tools or treatments that can have a significant impact on reducing the number of selfie deaths.

Reproducibility: More detailed analysis of the selfie deaths is shown on our web page[1], and our code and the dataset is also available for download.

5.1 Related Work

The trend and culture of posting selfies on social media have been investigated widely over the past few years. The popularity of selfies being posted on online social media has drawn a lot of researchers from different fields to study the various aspects of the selfie trend. We present the relevant work from major fields in this section.

The impact of selfies: Brager et al. studied the effect of a particular selfie on playing a part in a revolutionary movement [34]. The authors specifically analyzed death of a young teenager in Lebanon who died moments after taking a selfie near a golden SUV, that blew up. His death and the specific selfie stirred the Western news media and spectators, revolutionizing the movement - #NotAMartyr over the Internet. The authors argued that the practice of selfie-taking made the young boy's story legible as a subject of grievance for the Western social media audience. Porch et al. analyzed how the selfie trend has affected women's self-esteem, body esteem, physical appearance comparison score, and perception of self [232]. Baishya et al. found the effect

[1] http://labs.precog.iiitd.edu.in/killfie/

of selfies by candidate prime minister in Indian general elections was significant towards his victory [18]. Lim et al. suggested that insights into the selfie phenomenon can be understood from socio-historical, technological, social media, marketing, and ethical perspectives [174].

Psychology Studies: Qiu et al. analyzed the correlations between selfies and the personalities according to Big Five personality test of the participants [234]. Authors used signals such as camera height, lips position and the portrayed emotion to make predictions about their emotional positivity, openness, neuroticism and conscientiousness. Li et al. proposed that people taking selfies have narcissistic tendencies and the selfie-takers use selfies as a form of self-identification and expression. The role of selfies was also analyzed in making the selfie-taker a journalist who posts images on social media after witnessing events [146]. Senft et al. analyzed the role that selfies play in affecting the online users. It further shows how selfie as a medium has a narcissistic or negative effect on people [250].

Dangers of Selfie: An important theme, which is directly related to our paper is work related to the dangers that trend of selfie taking puts a selfie-taker in. Lakshmi et al. explain how the number of likes, comments and shares they get for their selfies are the social currency for the youth. The desire of getting more of this social currency prompts youth to extreme lengths [156]. Flaherty et al. [71] and Bhogesha et al. [23] talk about how selfies have been a risk during international travel. Howes et al. analyzed the selfie trends as a cultural practice in the contemporary world [126]. Authors particularly analyzed the case of spectators clicking selfies in the sport of cycling. The spectators wanted to capture the moment but ended up in obstructing the path of cyclists, leading to crashes. Subrahmanyam et al. work is the closest to ours discussing the dangers of taking a selfie [269]. Authors also provided statistical data about the number of deaths and injuries. A noble initiative #selfietodiefor[2] has been posting about the dangers of taking a selfie in a risky situation. They use Twitter handle @selfietodiefor for sending out awareness tweets and news stories related to selfie deaths.

Besides all the above-mentioned areas, researchers have also tried to distinguish selfies from other images by use of automated methods [41]. A project called *Selfie City* has been investigating the style of selfies in five cities across the world [189]. Using the dataset collected, they explored the age distribution, gender distribution, pose distribution and moods in all of the selfies collected. Researchers have also explored the use of nudging to alert a smart phone user about the possible privacy leaks [295], a technique which can readily be applied to warn users of the dangers of taking selfies in the present location/situation.

In this work, we study the dangerous impacts of clicking a selfie. Our work is the first in trying to characterize all the selfie deaths that have occurred in the past couple of years. Till now, there has been no research that proposes features and methods to identify dangerous and non-dangerous selfies posted on social media, which is what we propose to do in this work.

5.2 Selfie Deaths Characterization

In our work, we define a selfie-related casualty as *a death of an individual or a group of people that could have been avoided had the individual(s) not been taking a selfie*. This may even

[2] http://www.selfietodiefor.org/

involve the unfortunate death of other people who died while saving or being present with people who were clicking a selfie in a dangerous manner.

To be able to better understand the reasons behind selfie deaths, victims, and such incidents, we collected every news article reporting selfie deaths. We used a keyword based extensive web searching mechanism to identify these articles [268]. Further, we only considered those articles as credible sources which were hosted on the websites having either their Global Alexa ranking less than 5,000, or having a country specific Alexa rank less than 1,000. The earliest article reporting a selfie death that we were able to collect was published in March 2014. Two annotators manually annotated the articles to identify the country, the reason for death, the number of people who died, and the location where the selfie was being taken.

Country	**Number of Casualties (N=127)**
India	76
Pakistan	9
USA	8
Russia	6
Philippines, China	4
Spain	3
Indonesia, Portugal, Peru, Turkey	2
Romania, Australia, Mexico, South Africa, Italy, Serbia, Chile, Nepal, Hong Kong	1

Table 5.1: Country-wise number of selfie casualties

Using our approach, we were able to find 127 selfie-related deaths since March 2014. These deaths involved 24 group incidents, and others were individual incidents. By group incidents, it is meant that multiple deaths were reported in a single incident. An example of this could be an incident near Mangrul lake in the Kuhi district in India, where a group of 10 youth had gone for boating in the lake. While they were trying to take selfie, the boat tilted, and 7 people died. We count all such incidents as group incidents. Out of all the group incidents, 16 of the incidents involved 2 individuals, 5 involved 3 people, 1 incident had 5 casualties, and there were 2 group incidents claiming the lives of 7 people each. By analyzing selfie deaths - in terms of group and individual deaths, it can be concluded that taking dangerous selfies not only puts the selfie-taker at a risk but also can also be hazardous to the people around them. Although it is known that women take more selfies than men [189], however, our incident analysis showed that men are more prone to taking dangerous selfies, and accounted for roughly 75.5% of the casualties. Out of all the deaths, 41 victims were aged less than 20 years, 45 were between 20 and 24 years of age and 17 victims were 30 years old or above. This is consistent with our earlier finding that the trend of taking selfies is really popular among millennials.

Studying the geographic trends of the selfie deaths, we observed that India accounted for more than 51.76% of the overall incidents, out of which 87% were water-related casualties. In the USA, 3 deaths occurred while trying to click a selfie with a weapon, followed by Russia with 2 casualties. This might be a consequence of the open gun laws in both the countries. Distribution of incidents according to the country is shown in Table 5.1.

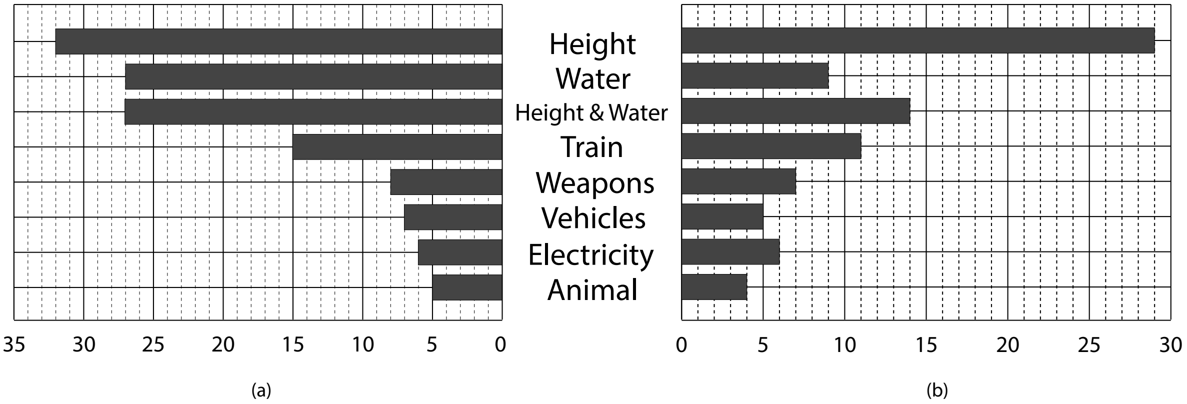


Figure 5.3: (a) Number of Deaths due to various reasons, and (b) Number of Incidents.

We looked at all the articles in our database to figure out what are the most common factors/reasons behind selfie deaths. Overall, we were able to find 8 unique reasons behind the deaths. We found that most common reason of selfie death was height-related. These involve people falling off buildings or mountains while trying to take dangerous selfies. Figure 5.3 shows the number of casualties for various reasons of selfie deaths. From the plot, it can be observed that for water-related causes, there were more group incidents. There were also considerable number of incidents where the selfie-taker exposed himself to both the height related and water body related dangers, thus we have analyzed such incidents separately. Twenty-seven individuals who died in 14 incidents qualified for this category. The second most popular category was being hit by trains. We found that taking selfies on train tracks is a trend. This trend caters to the belief that posting on or next to train tracks with their best friend is regarded as romantic and a sign of never-ending friendship.[3]

After analyzing selfie deaths, we can claim that a dangerous selfie is the one which can potentially trigger any of the above-mentioned reasons for selfie deaths. For instance, a selfie being taken on the peak of a mountain is dangerous as it exposes the selfie taker to the risk of falling down from a height. To be able to warn more users about the perils of taking dangerous selfies, it is essential to have a solution that can distinguish between the dangerous and non-dangerous selfies. Motivated by the reasons that we found for selfie deaths, we formulated features which would be ideal to provide enough differentiation between the 2 categories. In future sections, we discuss in detail as to how we generated features for different selfie-related risks and develop the classifier to identify selfies that are potentially dangerous.

5.3 Selfie Dataset Curation

We used *Twitter* for our data collection. Twitter is a popular social media website which allows access to the data posted by its users through APIs. Twitter provides an interface via its *Streaming*

[3] http://www.dw.com/en/dangerous-trend-the-train-track-selfie/a-18932440

API to enable researchers and developers to collect data.[4] Streaming API is used to extract tweets in real-time based on the query parameters like words in a tweet, location from where the tweet is posted and other attributes. The API provides 1% sample of the entire dataset [204]. We collected tweets related to selfies using keywords like *#selfie*, *#dangerousselfie*, *#extremeselfie*, *#letmetakeaselfie*, *#selfieoftheday*, and *#drivingselfie*.

We collected about 138K unique tweets by 78K unique users. The descriptive statistics of the data are given in Table 5.2.

Total Tweets	138,496
Total Users	78,236
Total Tweets with Images	91,059
Total Tweets with geo-location	9,444
Total Tweets with Text besides Hashtags	112,743
Time of first Tweet in our Dataset	Mon Aug 01
Time of last Tweet in our Dataset	Tue Sep 27

Table 5.2: Descriptive statistics of Dataset collected for Selfies

Out of the 138,496 tweets collected, we only found 91,059 to have images in them. We consider only those tweets for further analysis. However, it is not clear if all of those images were actually selfies or not. To retain only the true selfie images, we build a classifier based on image features to retain only the images that are selfies. We explain the classifier used below.

Preprocessing: We manually annotated 2,161 images as to determine whether they were selfies or not. Out of the tagged images, we found that 1,307 (roughly 60%) were selfies, and remaining 854 were not selfies. Using the manual annotations as ground truth, we constructed a classifier to discriminate between the selfies and non-selfies. The classifier was based on the transfer learning based model called DeCAF proposed by Donahue et al. [59]. DeCAF model first trains a deep convolutional model in fully supervised setting, and then various features from this network are extracted and tested on generic vision tasks. The deep convolutional model is as mentioned in Szegedy et al. [273]. The convolutional model has been trained and tested on the task of classifying 1.2 million images in ImageNet LSVRC - 2010 contest into 1,000 classes. It obtained top-1 and top-5 error rates as 21.2% and 5.6% respectively. As specified in the DeCAF framework, we use this trained model for the task of identifying if an image is a selfie image or not. This approach is useful as the cost of annotating all images as to whether it is a selfie or not is saved, and most convolutional deep learning models require enormous amounts of training data to train effectively from scratch. Therefore by using DeCAF, we built on the generic features provided by the original convolutional neural network. We found that algorithm gave 88.48% accuracy with 10-fold cross validation.

Using the model trained on the annotated dataset, we obtained labels for all of the non-annotated images. We found that out of 90K images (tweets with images or tweets hyper-linking to images), 62K were actually selfies. These 62K tweet set contained only 6,842 tweets which had a geolocation.

[4] `https://dev.twitter.com/streaming/overview`

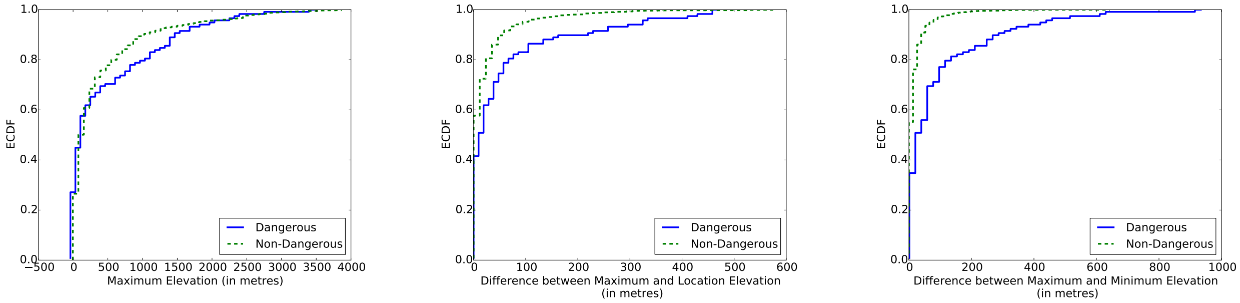


Figure 5.4: CDF Plots showing the difference in the distribution of height-related features for dangerous and non-dangerous images. Left: Maximum Elevation in 5km radius and 5 sampled locations (p-value:0.028). Center: Maximum difference in elevation of 10 points sampled in 1km radius with the elevation of the location (p-value: 7.09e-6). Right: Maximum Elevation Difference of 10 points sampled in 1km radius (p-value: 1.22e-9).

5.4 Feature Set Generation

In this section, we discuss the features we use for our classifier to differentiate between dangerous and non-dangerous selfies. Based on the analysis of selfie casualties we did in Section 5.2, we design different features for every major possible selfie-related risk (see Figure 5.3). We analyze each of the possible causes and consider what all features are possible in terms of tractability and availability. We first review the location-based features.

Height Related Risks: From our dataset, we observed that 29 selfie deaths were because of falling from an elevated location. We take this as an indication that taking selfies at an elevated location is dangerous. Based on the location of the selfie, we want to generate features that tell us if an image has been taken at an elevated location or not. To estimate the elevation of a location, we used Google Elevation API.[5]

Taking only the elevation of a particular place is not be informative to tell if the location is actually dangerous or not. For example, if a city is at a higher altitude, that does not make it necessarily dangerous. However, sudden changes in the nearby terrain indicate that there is a steep decrease in elevation, making the location dangerous. Google Elevation API returns negative values for certain locations such as water body. We formulated the following features based on the elevation of the location:

- *Elevation of the exact location of the selfie:* This feature was not informative as it captures only the elevation of the location, and that does not necessarily mean a risk due to height. This was validated by the fact that p-value of Kolmogorov-Smirnov (KS) 2 sampled test was 0.12; which we can reject only in 15% confidence interval.
- *Maximum Elevation of the surrounding area:* To get a sense of the area surrounding the exact location, we sample 10 locations in 1-km radius and return the maximum elevation out of those. We choose the specified value of radius and number of locations because

[5] https://developers.google.com/maps/documentation/elevation/

they returned the lowest p-value after applying 2-sample KS test for dangerous and non-dangerous selfie distribution.

- *Difference Elevation of the surrounding area:* We calculate this as the maximum difference between the elevation of our exact location and the sampled locations' elevation. These features capture the sudden elevation drop that might exist near the surrounding area. For this feature, we sampled 5 locations in a 5-km radius for the same reason as mentioned above.
- *Maximum Elevation Difference in the surrounding area:* Taking the maximum difference between the highest elevation and lowest elevation of the sampled points helped us capture the amount of elevation variation in the surrounding area.

We did not work with other possible statistics such as the average elevation or median elevation as those statistics try to capture the center point or a single representative value of the distribution. We are however interested in sudden elevation drops in the surrounding area, which will lie on the extremes of the elevation distribution.

To evaluate the efficiency (or the discriminative power) of the above-mentioned features, we plot the empirical cumulative distributions (CDF) of height-related dangerous selfies and non-dangerous selfies. This can be seen in Figure 5.4. We can notice that for the 3 features, the empirical CDF of dangerous and non-dangerous selfies are considerably different. The KS test returned p-values:0.028 for Maximum elevation, 7.09e-6 for Elevation difference between maximum elevation and our location and 1.22e-9 for Maximum elevation difference.

Water Related Risks: Another prominent reason of selfie casualties that we infer from Figure 5.3 is water-related risks. After analyzing the water-related incidents, we found that often people took selfies while being in a water body or in close proximity to one. They ended up drowning by losing their body balance and falling into the water body. To tackle water related risks, we generate features based on the proximity of their location to a water body. Consider the selfie in Figure 5.5(a) which has been taken in the middle of a water body. We mapped the exact location of the selfie to Google Maps and considered 500×500 pixel image pertaining to level 13 zoom factor on Google Maps [7]. The image after this step looked like in Figure 5.5(b). We applied image segmentation to identify the contour of all the water bodies shown in Figure 5.5(c). To infer whether a given location is in close proximity to a water body or not, we use the minimum distance to a water body from the location of the image as a feature. Since all the segmented images were of maps with same scale and zoom factor, the distance was treated as pixel location distance. Proximity to a small water body like a stream or a river might not make a selfie dangerous, therefore we also use fraction of the pixels in the segmented image (Figure 5.5(c)) to further help us in distinguishing between dangerous and non-dangerous selfies.

We can observe from the Figure 5.6 that for both of the water features - minimum distance to a water body and the fraction of water pixels in the segmented image, the distribution of water-related dangerous and non-dangerous selfies is considerably different. We use 2-sampled KS test to statistically confirm our observations. We obtained p-values of $1.18e$-19 (minimum distance to a water body) and $2.79e$-19 (fraction of water pixels in the segmented image) indicating that we can safely reject that the features are being generated from the same distribution.

Train/ Railway Related Risks: Besides water and height-related risks, another common reason of selfie casualties is train-related risks which accounted for 11 casualties. We used Google

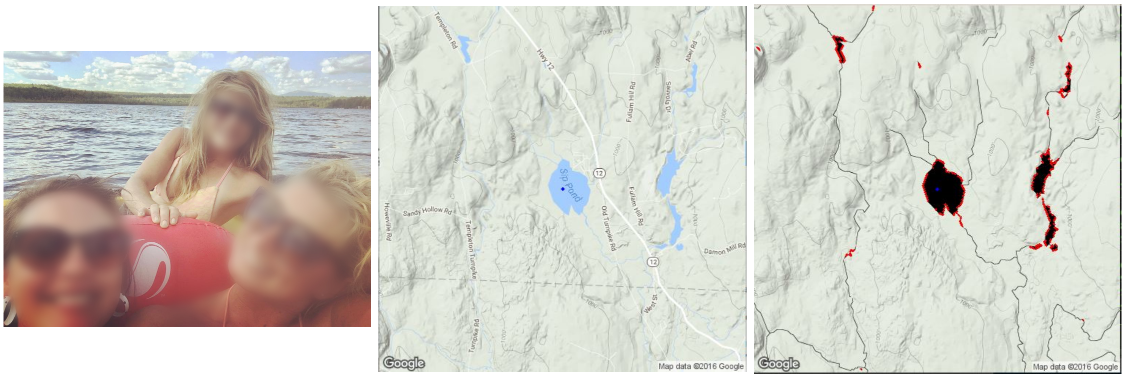


Figure 5.5: Segmentation Example: Different stages of processing to get the final segmented image distinguishing between the water and land.

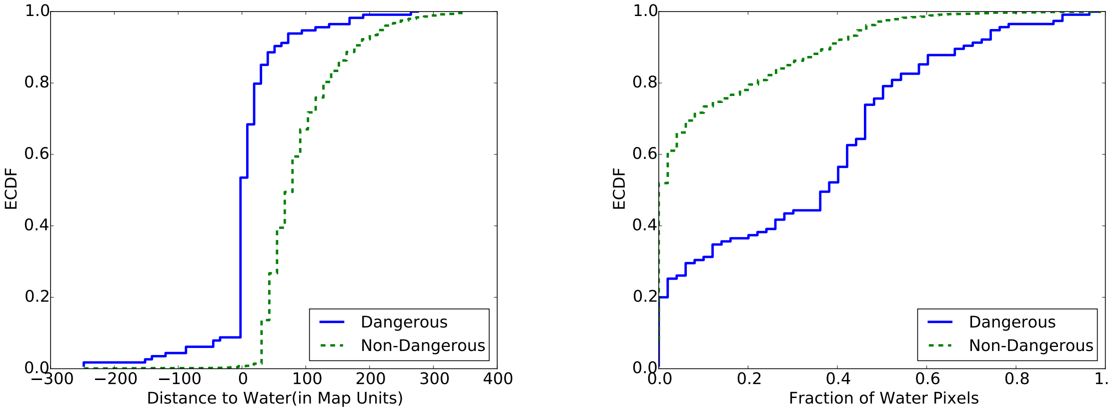


Figure 5.6: CDF Plots showing the difference in dangerous and non-dangerous distributions for water-related features. Left: Minimum distance to a water body. Right: Fraction of water pixels in the segmented image

Places API to determine if there is a railway track or a railway station close to the location of the selfie or not. We used the minimum distance between the location and the railway track as a feature. Though this feature is not sufficient to distinguish between dangerous and non-dangerous selfie, it still provides valuable information which when appended to other features proves to be helpful in the classification task.

Driving/Road Related Risks: It is challenging to account for driving-related risks in all possible contexts. The location of the selfie can provide information about how close a person is to a road. Using only the location data is not sufficient to determine if the selfie-taker was driving at the time of taking a selfie, or was standing in the middle of a busy road to take the selfie. However, we still think that the minimum distance of the location of the selfie to the highway/road will be informative in determining the 'dangerousness' of the selfie when used in conjuction with other features.

For all the other reasons such as weapons, animal, electricity, it is difficult to find location based insights, and thus impossible to find location based features. We rely on other signals based on the text accompanying the selfie, and the content of the image to be able to derive features which can provide insights about these reasons. For example, the presence of a weapon or animal can be easily inferred from the image content. Below, we discuss the text-based and image content-based features.

Text-based Features: The content of the tweet can be a useful source for indicating if the image accompanying it is a dangerous selfie. Users tend to provide context to the image either directly in the tweet text or through hashtags. We use both to generate our text-based features. After removing the URLs, tokenizing the tweet content, and processing emojis, we obtain our text input. We use TF-IDF over the set of unigrams and bigrams. For further enriching the text feature space, we convert the text into a lower dimension embedded vector obtained using *doc2vec*[167].

Image-based Features: Since an image could be dangerous due to various reasons, we cannot simply apply a classifier to the actual pixels of the image. Classifying an image as to whether it is dangerous or not requires more understanding of the context and the elements in the image. Therefore, we first extract the salient regions in images and then generate captions for each of those regions.

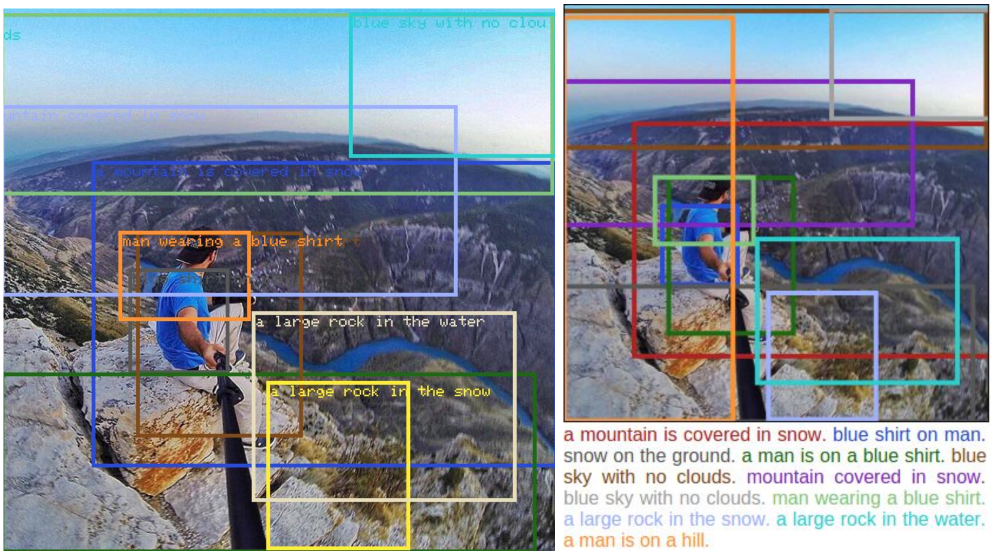

blue sky with no clou
ds
a mountain is covered in snow
man wearing a blue shirt
a large rock in the water
a large rock in the snow
a mountain is covered in snow. blue shirt on man. snow on the ground. a man is on a blue shirt. blue sky with no clouds. mountain covered in snow. blue sky with no clouds. man wearing a blue shirt. a large rock in the snow. a large rock in the water. a man is on a hill.

Figure 5.7: An example of the DenseCap on one of the images (Left) from our dataset. We use the dense captions produced by DenseCap (Right) to come up with text based features over them.

To extract informative regions in images and for the caption-generating process, we used DenseCap [133]. DenseCap is start-of-the-art deep learning based captioning technique for regions in an image. It outperforms other models such as Full Image RNN, Region RNN on both tasks of dense captioning and as well as image retrieval comfortably. The average precision on the dense captioning task by DenseCap was 5.24, way higher than the closest competitor 4.88. The architecture of DenseCap involves a fully convolutional layer, a fully convolutional localization layer used for extracting ROI (regions of interest) and their features, a recognition network for finding relevant ROI's, and a language model to generate captions for the ROI. An example of the output of the DenseCap on a selfie in our dataset is shown in Figure 5.7.

We treat the generated captions as the text describing the image in natural language. From the text, we compute natural language features such as unigrams, bigrams to determine if the content of the image is dangerous or not. We also convert the captions generated into a lower dimension vector in a similar fashion we did for text-based features. To empirically view the validity of our approach, we plotted the 2-dimensional t-SNE (Stochastic Neighbor Embedding) [186] mapping of the embedded *doc2vec* vectors in Figure 5.8. In the plot, we can see that the triangles (dangerous selfies) are negative in the 1st vector components (X-axis), whereas the circles (non-dangerous selfies) are largely positive. On the plot, we can imagine a line easily separating most of the dangerous and non-dangerous selfies. Our entire feature space could be categorized as shown in Table 5.3.

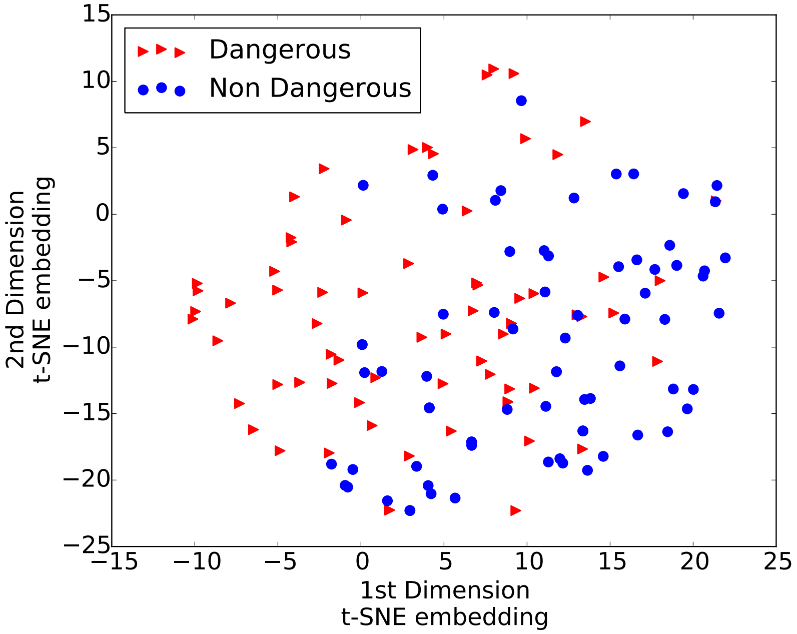


Figure 5.8: t-SNE scatter plot of doc2vec output of generated captions for 50 randomly chosen dangerous and non-dangerous selfies.

5.5 Experiment

5.5.1 Manual Annotation

From the selfie data set described in Section 5.3, we sampled a random set of 3,155 selfies with geolocation for creating an annotated data set. We manually labeled the images to determine whether they are dangerous or not. For the process of annotations, we asked questions such as, whether the image depicted is dangerous or not? If yes, then what is the possible reason for it being dangerous? And, whether text accompanying the image helped them in classifying if

Table 5.3: Location-based, Image-based and Text-based features used for classification of selfies.

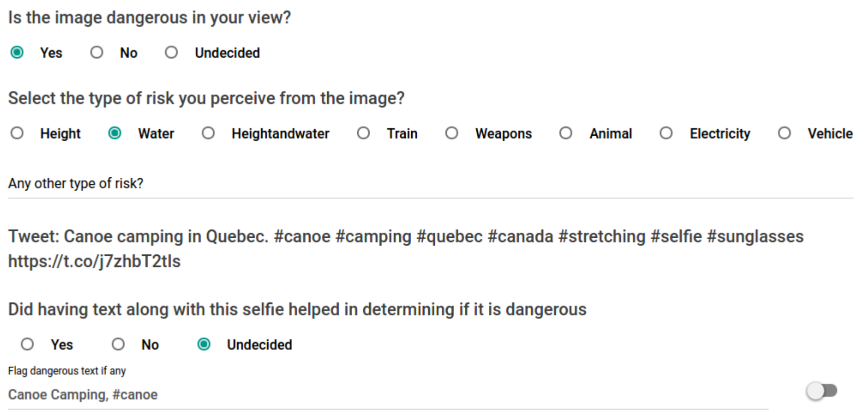
Feature Type	**Feature**
Location Based Features	Elevation of the location Maximum Elevation Difference between Maximum elevation out of sampled points and elevation of the location. Maximum elevation difference in the set of sampled points Minimum Distance to water body Fraction of water pixels in the segmented image Distance to railway tracks Distance to major roadway/highway
Image Based Features	TF-IDF of unigrams and bigrams on DenseCap captions Doc2Vec representation of DenseCap captions
Text Based Features	TF-IDF of unigrams and bigrams on the Twitter text Doc2Vec representation of Twitter text

image is dangerous or not, and so on. A screenshot of the tool is shown in Figure 5.9.[6] We asked 8 annotators to annotate the set of 3,155 selfies, randomly split into a common set having 400 images. The common set was annotated by every annotator, and the shared set was divided equally among all the annotators. The inter-annotator agreement rate obtained on the common set of 400 selfies, using the Fleiss Kappa metric [72] was 0.74. Fleiss kappa metric interpretation reveals that the above value indicates substantial agreement between the annotators [165]. The annotated dataset contained 396 dangerous and 2,676 non-dangerous selfies. Annotators were unsure about the remaining selfies in our dataset. For the annotated images, we found that vehicle related causes for a selfie being dangerous, like taking a selfie in a car, is the maximum, followed by water related risks. Statistics about the risks that annotators perceived from the dangerous images is given in Table 5.4. Annotators frequently found images to be dangerous in more than one aspect. For such cases, we counted their labels for all the mentioned risk types. One striking observation is that even though we didn't find any selfie casualties due to road related incidents in our research, it was identified as a potential risk by the annotators in as many as 29 dangerous images (7%).

5.5.2 Classifier

Considering the annotations performed in the section above as ground truth, we evaluate the performance of our classifier on the task of classifying a selfie as dangerous. The problem of classifying dangerous selfies is a highly unbalanced problem. We have only 623 (roughly 9%) dangerous selfies in comparison to the remaining 5,837 non-dangerous selfies. The imbalance in the annotated data is a common problem in many machine learning applications. In these cases, applying a classifier on the data as is, leads to a classification algorithm to simply predict the majority class label for all the samples. To avoid this, many methods have been proposed

[6] The annotation tool we used is available at http://twitdigest.iiitd.edu.in:4000



Is the image dangerous in your view?

● Yes ○ No ○ Undecided

Select the type of risk you perceive from the image?

○ Height ● Water ○ Heightandwater ○ Train ○ Weapons ○ Animal ○ Electricity ○ Vehicle

Any other type of risk?

Tweet: Canoe camping in Quebec. #canoe #camping #quebec #canada #stretching #selfie #sunglasses https://t.co/j7zhbT2tls

Did having text along with this selfie helped in determining if it is dangerous

○ Yes ○ No ● Undecided

Flag dangerous text if any

Canoe Camping, #canoe

Figure 5.9: Screenshot of the annotation tool. We asked above questions to the annotators based on a selfie image shown to them.

Table 5.4: Reasons marked by annotators for a selfie being dangerous.

Reason	**Number of Dangerous Selfies**
Vehicle Related	120
Water Related	118
Height Related	86
Height and Water Related	55
Road Related	29
Animal Related	16
Train Related	8
Weapons Related	4

in the literature for balancing such data sets [113]. For our task, we experimented with random down-sampling (randomly removing samples from majority class).

As mentioned earlier, our feature space can be easily divided into 3 categories - text, image, and location-based. To compare all feature types, we build and test the classifiers for every possible combination of the features. For all our experiments, we perform 10-fold cross validation. Furthermore, we use the grid search to find the ideal set of hyperparameters for each classifier by doing a 3-fold cross-validation on the training set. We tested the performance of our method using 4 different classification algorithms - Random Forests, Nearest Neighbors, SVM and Decision Trees. Each of the classifiers was trained and tested on the similar dataset and using the same feature configuration. Table 5.5 lists the accuracy obtained by using various classification techniques over different combinations of our feature space.

All the three features, when combined perform the best. This is also observed from the Receiver Operator Characteristic (ROC) graph, shown in Figure 5.10. To obtain this curve, we computed the probability of selfie being dangerous according to the best performing SVM model. For every selfie, we only took the probability when the particular selfie was in the testing-set when we were performing cross-fold validation. After sorting these probabilities, we generated equi-spaced 300 threshold points between $[0, 1]$, and marked any selfie above the threshold as dangerous and rest non-dangerous. When the assigned labels were compared with the annotated ground truth, we obtained true positive count and false positive count. All the feature permutations perform much better than the random baseline.

Table 5.5: Average accuracy (with standard deviation) for 10-fold cross validation over different classification techniques and different feature configurations for the down sampled dataset.

	SVM	RandomForest	Nearest Neighbors	Decision Tree
Image Only	$\mathbf{0.795 \pm 0.044}$	0.774 ± 0.048	0.547±0.034	0.741 ± 0.039
Text Only	$\mathbf{0.649 \pm 0.047}$	0.533 ± 0.039	0.514 ± 0.017	0.536 ± 0.036
Location Only	0.638 ± 0.037	$\mathbf{0.639 \pm 0.033}$	0.596 ± 0.055	0.625 ± 0.034
Image + Location	$\mathbf{0.794 \pm 0.039}$	0.773 ± 0.023	0.559 ± 0.031	0.738 ± 0.040
Text + Location	$\mathbf{0.679 \pm 0.018}$	0.607 ± 0.041	0.51 ± 0.04	0.599 ± 0.03
Text + Image	$\mathbf{0.811 \pm 0.016}$	0.767 ± 0.031	0.566 ± 0.039	0.765 ± 0.054
Text + Image + Location	$\mathbf{0.824 \pm 0.039}$	0.779 ± 0.035	0.574 ± 0.032	0.750 ± 0.03

Multimodal features are important: Based on the results in Table 5.5, we can observe that when all the classes of features are used, the accuracy is the highest. This validates our approach of using multimodal features. It can also be seen that the combination of image and text features perform better than the image and location features as seen by [43]. This might be indicative that the context and the content of the selfies is a far better predictor than the location of the selfie.

Image features perform well: Further analyzing the results, we can clearly see that image-based features perform the best out of all the classes of features. Therefore, even in the absence of location of the selfie, a model based only on the image based features can perform relatively well in finding dangerous selfies. This can be helpful in cases, where the user's post is not geocoded, or in an application case when location information is not available due to GPS being turned off or unavailable.

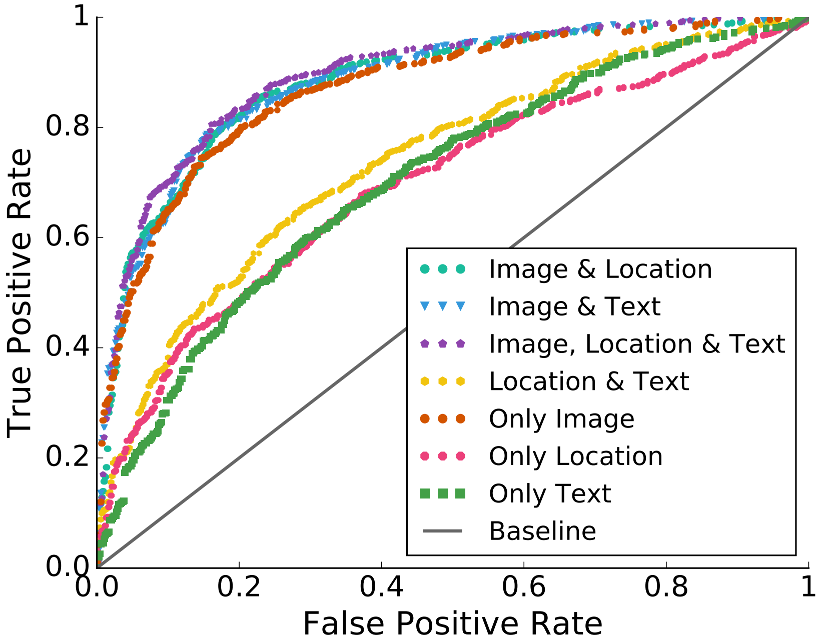


Figure 5.10: Receiver Operating Characteristic (ROC) curves corresponding to the statistical models for identifying dangerous selfies. “Dangerous” selfie is the positive class.

5.6 Conclusions

In this paper, we create a novel dataset of reported selfie casualties to describe the subtleties of the situations where such accidents may occur. Our work demonstrates the viability of using selfies and content posted on Twitter as an instrument to quantify and characterize *dangerous selfies* that may cause casualty to selfie-ers. Further, we present a multimodal classifier that uses various features such as - text-, image-, and location-based features to identify dangerous selfies. In this work, we demonstrate that measuring the multimodal subtleties (image, text, and location) of selfie tweets available on social media can help to identify physical harm possibilities to selfie-ers. We show that location-based features can be customized to detect the common reasons such as water-related, height-related factors pertaining to selfie casualties. We adopt state of the art deep learning techniques such as DenseCap to determine the content of the selfie. The approach demonstrated in our work, suggests that even in absence of one or more of the above mentioned features, technologies can be developed to identify dangerous selfies. We believe that there is an opportunity to extend our approach for identifying selfie-ers who are at high risk of selfie-related casualties.

Limitations: Our work explores a set of Twitter users, who are explicit about sharing selfies and mention hashtags such as #selfies and #myselfie in their posts. However, we acknowledge that these users may not be representative of the entire Twitter or general social media population. There could be a section of users who may not be explicit about sharing selfies using hashtags or keywords. We also acknowledge, that there may be a section of selfie-ers who may not be sharing their selfies on social media. There might be an inherent selection bias towards selfie-ers who prefer to use Twitter as a platform to share selfies.

Hemank Lamba et al. ”Driving the Last Mile: Characterizing and Understanding Distracted Driving Posts on Social Networks”. Fourteenth International Conference on Web and Social Media (ICWSM) 2020.

CHAPTER 6

DETECTING DISTRACTED DRIVING POSTS ON SOCIAL MEDIA

In 2015, 391,000 people were injured due to distracted driving in the US. One of the major reasons behind distracted driving is the use of cell-phones, accounting for 14% of fatal crashes. Social media applications have enabled users to stay connected, however, the use of such applications while driving could have serious repercussions - often leading the user to be distracted from the road and ending up in an accident. In the context of impression management, it has been discovered that individuals often take a risk (such as teens smoking cigarettes, indulging in narcotics, and participating in unsafe sex) to improve their social standing. Therefore, viewing the phenomena of posting distracted driving posts under the lens of self-presentation, it can be hypothesized that users often indulge in risk-taking behavior on social media to improve their impression among their peers. In this paper, we first try to understand the severity of such social-media-based distractions by analyzing the content posted on a popular social media site where the user is driving and is also simultaneously creating content. To this end, we build a deep learning classifier to identify publicly posted content on social media that involves the user driving. Furthermore, a framework proposed to understand factors behind voluntary risk-taking activity observes that younger individuals are more willing to perform such activities, and men (as opposed to women) are more inclined to take risks. Grounding our observations in this framework, we test these hypotheses on 173 cities across the world. We conduct spatial and temporal analysis on a city-level and understand how distracted driving content posting behavior changes due to varied demographics. We discover that the factors put forth by the framework are significant in estimating the extent of such behavior.

Distracted driving is any non-driving activity that the driver engages in, which can lead to visual (taking eyes off the road), manual (taking hands off the driving wheel) or cognitive (taking the mind off driving) distractions [218]. Distracted driving is particularly risky: In 2015, fatal crashes involving distracted drivers resulted in the deaths of 9 individuals and 1,000 injuries in the US alone [216].

Usage of cell-phones while driving has been a primary reason for distraction-affected crashes, resulting in 69,000 total crashes in 2015 [216]. Texting while driving can be particularly devastating as it combines all three types of distractions (visual, manual, and cognitive) [39, 125, 177, 289]. Among cell-phone users, teenagers and young adults are especially at risk. Studies show that 42% of high schoolers text multiple times while driving [137], and teenagers and young adults comprise 36% of distracted drivers using cell phones [216].

We argue that social media use can have similar effects. Individuals spend 30% of their weekly online time on social networking applications [85], with 78% of traffic coming from smartphones [219]. For instance, an average Snapchat user spends 30 minutes daily on the platform [263]. However, while many studies investigated the risk of using cell phones while driving, prior work generally focused on texting and emailing; thus, the impact of social media use remains relatively unknown.

We address this gap in our paper, by using large-scale data from Snapchat to develop a deep-learning based classifier to classify a post as distracted driving content or not. Then, grounded in Lyng's edgework theory [184] (details in the next section), we investigate the extent to which people create and post content while driving or while being in the passenger seat, [1] and characterize the users and spatial and temporal patterns associated with higher incidence of such content.

We discover that (1) a deep learning classifier trained on content has a good performance in detecting distracted driving content, (2) distracted driving content posting behavior is widespread - 23% of snaps posted are related to distracted driving. Further, by analyzing the spatial and temporal patterns, we discovered that (3) distracted driving content is generally posted in night-time and regional affects are visible in the temporal patterns of such behavior and (4) distracted driving content posts are concentrated to only certain spots in the city. Finally, we also discovered that age and gender play a key role in inferring who is more likely to participate in such risk-taking behavior.

In summary, we make the following main contributions: (1) a classifier to detect distracted driving content posting behavior on Snapchat; (2) an empirical study characterizing the extent of distracted driving content behavior across 173 cities around the world, the types of users more likely to engage in such behavior, and spatial and temporal patterns of distracted driving content snaps in these cities.

Our results have implications for platform designers and policymakers. Our proposed deep-learning based classifier can identify distracted driving content content posted on social media. Furthermore, the spatial and temporal patterns and individual user characteristics we uncover can inform the design of region-specific interventions for certain cities where such behavior is common, and for specific times when users generate these posts; as well as the design of individual-level interventions and educational campaigns for at-risk populations.

[1] Arguably a front-seat passenger creating social media content, e.g., a video, could also be a source of distraction for the driver.

6.1 Development of Research Questions

Our work is grounded in two theoretical frameworks. First, Goffman's dramaturgical theory [86] describes how individuals may engage in risk-taking behavior to improve their peers' impressions of them, even when interacting through online social media platforms [122]. Goffman introduced the term "impression management", which has been widely used to explain how an individual presents an idealized rather than a more authentic version of themselves [86]. In the context of risk-taking behavior, Leary et al. [168] analyzed voluntary risk-taking activities such as avoiding condoms, indulging in narcotics and steroid use, and reckless driving, and suggested that such risk-taking activities are undertaken to improve the impression of individuals among their peers [168]. Hogan [122] extended Goffman's concept of impression management to online social media websites and considered the online social media platforms as a stage that allows users to control their impressions via status messages, pictures posted, and social media profiles. Similarly, we expect that social media users could post distracted driving content. Therefore, we ask our first research question:

RQ1. [Extent] *What is the extent of distracted driving content posting behavior on Snapchat?*

Second, Lyng's edgework theory [184] characterizes voluntary risk-taking behavior (or, edgework) and identifies a range of individual and social factors that characterize the edgeworkers. The framework defines edgework activities as those where there is a "clearly observable threat to one's physical or mental well-being", such as rock-climbing, auto-racing, criminal behavior, drug use, etc. Edgework theory is social psychological, resting on the idea that individuals indulge in such activity to maintain the "illusion of control." Treating illusory sense of control as a factor, Lyng observed that edgework is more common among young people than among older people and among males than females. Other studies have found similar evidence related to the gender and age of the risk-takers [63, 168]. Building on this line of work, we also investigate if the demographic factors put forward by edgework framework also hold for distracted driving content posting behavior on Snapchat. We therefore ask:

RQ2. [Demographics] *Which user demographic characteristics correlate with posting distracted driving content?*

Besides the individual characteristics, Lyng also noted that individuals who are under pressure from external social forces are also more inclined to do edgework, as a way to exhibit control over experiences that are potentially even more dangerous. We expect that different geographic locations can give indications about the culture in that particular part of the world and hence the social forces at play. In addition, social media use is known to vary across geographies [120, 140, 280]. For example, Kim et al. [140] studied how cultural contexts influence usage of social network sites among teenagers from US and Korea, finding that Korean participants used it for receiving acceptance from their peers, while US participants used the websites only for entertainment purposes. Similar studies were carried out by Hochman et al. [120] and Tifentale et al. [280], where they noticed different patterns across geographies in terms of photo-sharing behavior. A better understanding of the geographic patterns can help in designing more appropriate and effective interventions for the at-risk population in such regions. We, thus ask:

RQ3. [Spatial Analysis] *How does distracted driving content posting behavior vary across cities worldwide?*

There is much variability in the temporal patterns of social media usage. For example, Golder et al. [89] analyzed Facebook messaging pattern across universities and discovered temporal rhythms. They showed that students across all universities followed a "weekday" and a "weekend" pattern and further showed that students in the same university behaved similarly. Grinberg et al. [97] discovered interpretable temporal patterns for mention of different terms related to nightlife, coffee, etc. on Twitter and Foursquare checkins. Golder et al. [88] further analyzed the temporal patterns of Twitter messages and were able to identify diurnal and seasonal mood rhythms, such as observing that people were generally happier on weekends; and that the morning peak in the number of messages was delayed by 2 hours on weekends. We investigate whether we can derive similar diurnal patterns for distracted driving content posting behavior, and ask:

RQ4. [Temporal Analysis] *How does distracted driving content posting behavior vary with time?*

However, before we can begin to study distracted driving content posting behavior on Snapchat empirically, we first need to be able to detect such behavior. A major component of our work building a classifier to identify distracted driving content , where the content creator is driving or is distracted while driving. A popular stream of work in the area of classifying videos is to apply multiple image-based classifiers on the frames of the given video. To this end, He et al. [116] proposed a deep learning model that learns the residual functions and out-perform previous competitors in a widely popular ImageNet challenge. Zagoruyko et al. [311] further improved the ResNet model and proposed a Wide Residual Network (WRN), which uses the increased width of the network to improve accuracy. Xie et al. [303] modified the ResNet model by introducing a new hyper-parameter called cardinality to better tune the depth and width of the model. We use some of these architectures as candidate models for our deep learning classifiers. Among the video classification approaches used for action recognition, an approach that operates on spatio-temporal 3D CNNs stands out [109] by having high accuracy on standard action recognition datasets such as Kinetics and UCF101. Based on the above insights, we explore the feasibility of learning a robust classifier to distinguish between distracted driving content and non-distracted driving content, asking:

RQ5. [Detection] *How can we use Snapchat content to distinguish between distracted driving and other videos? Moreover, how accurate is such a classifier?*

6.2 Data Collection and Dataset

In this work, we study a widely used social media platform, Snapchat. Snapchat is a popular platform that allows users to post multimedia content(*snaps*) that can be shared with other users - visible by all or only by friends. Our dataset is based on SnapMap - a unique feature where any content can be posted publicly anonymously. The content posted on Snap Map is automatically geo-tagged and is shown in a localized region, though not giving the exact location.

[2] https://www.cia.gov/library/publications/the-world-factbook/appendix/appendix-b.html

Table 6.1: A sub-sample of the cities selected for analysis.

City	Economic Status[2]	Pop.	Male (% age)	Pop.(<20)
Cape Town	Developed	4.43M	48.90	0.329
London	Developed	9.05M	49.80	0.247
Melbourne	Developed	4.77M	49.00	0.241
New York	Developed	8.58M	47.70	0.232
Rio De Janeiro	Developing	13.29M	46.80	0.267
Riyadh	Developing	6.91M	59.17	0.220

6.2.1 Data Collection

For obtaining the data through SnapMap, we leverage the underlying API to collect data across 173 cities. We select these cities such that they give us a wide coverage over the entire world and they were constrained on having a minimum population of $200k$ each. Further, we filter out cities where there is limited or restricted Snapchat usage (for example, Chinese metropolises). A sampled list of some of the cities selected for this analysis, with certain attributes (that we use for future analysis) is provided in Table 6.1. We utilize the shapefiles obtained from OpenStreetMap[3] to precisely define the region enclosed by a city. In the absence of a city's shapefile, we use its bounding box values instead.

This overall city's region/bounding box is divided into smaller tiles using a grid such that each tile is $1km \times 1km$. A similar approach has been previously used in geographical studies on Snapchat [135] which utilizes a tile of size $2.4km \times 2.4$km respectively. We periodically collect snaps posted in each of these grid tiles, crawling each city once every 8 hours. The data collected lists the time at which the snap was posted in Coordinated Universal Time (UTC), which we then convert to the local time-zone of the corresponding city to allow for uniformity in the temporal analysis.

Table 6.2: Brief description of the data collected.

Number of Snaps collected	6,431,553
Number of cities scraped	173
Time of first Snap	16-03-2019 00:00:00
Time of last Snap	15-04-2019 23:38:57
Most active city	Riyadh (1,023,836)
Least active city	Havana (114)
Most active day	13th April, 2019 ($288K$)
Least active day	30th March, 2019 ($89K$)
% Snaps deleted	2.98%

Overall, a brief statistics of the collected dataset is given in Table 6.2. We observed that

[3] https://www.openstreetmap.org

204,874 snaps were deleted after posting and were not used in our analysis. Though our work is concentrated on Snapchat, it can be easily extended to most social media platforms where users post multimedia content (images/videos).

6.3 Detecting Distracted Driving Content

To be able to build a classification model, we need to have a ground truth dataset of snaps with labels marking each as either distracted driving content or non-distracted driving content. We built an annotation portal (details of the portal provided in Supplementary), and asked annotators to provide labels for over 15K snaps, randomly sampled from our dataset. We annotate each snap for distracted driving content or non-distracted driving content and ensure that at least three annotators annotated each snap. We obtained a Fleiss-Kappa inter-annotator agreement rate of 0.85, which signifies almost perfect agreement [73]. A snap was assigned a ground-truth label of distracted driving content if two or more annotators agree that it is a distracted driving content snap. An anonymized example of distracted driving content snap can be viewed at `https://rebrand.ly/driving-snap`. This snap is clearly dangerous as it is created by an individual who is driving and hence is classified as an example of distracted driving.

Dataset. We randomly sample and split the manually annotated snaps into training and test set of 8,634 (6,392 negative, 2,242 positive) and 1,479 snaps (1,118 negative, 361 positive) respectively. We train our model using 5 fold cross-validation on this so obtained dataset. The number of positive samples (distracted driving) in our training dataset is much less than the number of negative samples (non-distracted driving) which creates a class imbalance.

We experiment with two different kinds of classifiers - image-based and video-based. The main distinction between both types of approaches is that the image-based classifiers first converts the snap (a video) into frames, and then each frame is classified independently as either distracted driving content or non-distracted driving content. Post classification of each frame, various aggregation techniques (single and majority voting) are used to obtain a single label for the entire snap. On the other hand, the video-based classification methods use the entire video as

Table 6.3: Performance of various classification methods, using different base architectures on our ground truth dataset.

Type	Architecture	Accuracy	Precision	Recall	F1 Score
Image-Based (Single Voting)	ResNeXt-50	0.924 ± 0.005	0.780 ± 0.015	0.958 ± 0.01	0.859 ± 0.007
	ResNet-34	0.919 ± 0.007	0.774 ± 0.02	0.948 ± 0.013	0.851 ± 0.009
	WideResNet	**0.926 ± 0.009**	**0.792 ± 0.030**	**0.948 ± 0.016**	**0.862 ± 0.012**
Image-Based (Majority Voting)	ResNeXt-50	0.947 ± 0.001	0.902 ± 0.008	0.876 ± 0.011	0.888 ± 0.004
	ResNet-34	0.942 ± 0.004	0.896 ± 0.02	0.860 ± 0.012	0.877 ± 0.005
	WideResNet	**0.947 ± 0.003**	**0.914 ± 0.011**	**0.868 ± 0.019**	**0.890 ± 0.006**
Video-Based	ResNet-34	0.930 ± 0.008	0.860 ± 0.044	0.860 ± 0.033	0.857 ± 0.013
	ResNeXt-101	**0.941 ± 0.003**	**0.876 ± 0.015**	**0.880 ± 0.026**	**0.876 ± 0.008**

an input.

Image Based Methods. We build our image-based methods over existing image-based deep learning architectures. We leverage the best-performing classifiers that have achieved high accuracy on ImageNet Large Scale Visual Recognition Challenge [244]. The challenge consisted of 1.2M images covering 1,000 classes. Specifically, we experiment with ResNet-34 [116] (24.19% top 1 error), ResNeXt-50 [303] (22.2% top 1 error) and WideResNet-50 (WRN) [311] (21.9% top 1 error). The wide residual networks perform well as they decrease the depth of the network and increase its width to increase the representational power of the residual blocks. We pre-train these architectures on the ImageNet dataset, following which we fine-tune them on our annotated dataset using transfer learning. Such a technique is based on transfer learning and is efficient even when a small number of samples are used to fine-tune [312]. The number of training samples in our dataset after converting the videos to frames is 69,125, which is sufficient for transfer learning. To solve the class imbalance issue, we use data augmentation techniques such as random cropping and horizontal flipping to increase the number of driving frames shown to the network during training. For converting the snaps (videos) to frames, we sample a frame every second - every 30th frame per second (video's original playback rate is 30fps). For each frame, we obtain a label of whether it is distracted driving content or non-distracted driving content. To obtain a single label for the entire snap, we use two aggregation techniques - (a) Majority voting and (b) Single voting. For majority voting, we classify the entire snap to be distracted driving content if the majority of the frames are assigned to distracted driving class, whereas for single voting, we classify the entire snap as distracted driving content content if we classify even a single frame as distracted driving content. We tune the hyper-parameters of these models using 5-fold cross-validation and report their accuracy, precision, recall and F1 score [4] on the test set in Table 6.3.

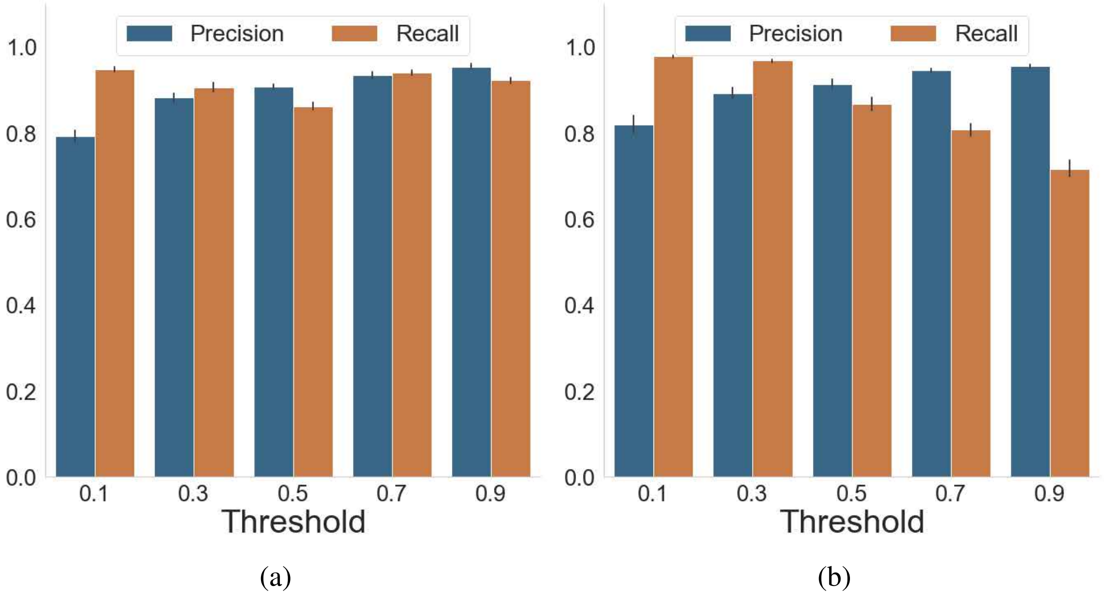
To measure the robustness of the frame selection, we compare our frame sampling strategy with that of a random frame sampling every second. We discover that the random frame sampling-based approach performs worse than our frame sampling strategy (random sampling has 93.8%, compared to our frame sampling's 94.8% accuracy). Similarly, we also experiment with different voting aggregation techniques - where a snap is assigned a label if more than 10%, 30%, 50%, 70% and 90% of the frames have the same label. We report these results in Figure 6.1.

Video Based Methods. For video-based classifiers, we again use state of the art architectures for a video classification task. Karpathy et al. explored multiple ways to fuse temporal information from consecutive frames using 2D pre-trained convolutions [138]. Similarly, Hara et al. proposed spatiotemporal 3D CNNs for video classification [109]. They examined deep architectures based on 3D Res-Net backbones for several datasets, achieving a top-5 accuracy of 85.7% on the Kinetics dataset[139]. The Kinetics dataset consists of more than 300K videos with 400 class labels. To adapt these architectures for our classification task, we re-train two of their pre-trained models, which are based on ResNet34 and ResNeXt-101 architectures over our annotated dataset. Similar to the image-based methods, we utilize random cropping to solve the class imbalance issue.

The image classifiers perform better than the video classifiers, as shown in Table 6.3. We hypothesize that this might be because the image classifiers are pre-trained on the ImageNet

[4] We report the precision, recall and F1 score of the minor class in all our results

dataset, which allows the classifiers to gain a much better internal representation of outdoor driving scenes. On the other hand, the Kinetics dataset on which the video classifier is pre-trained contains labels for action recognition tasks which do not transfer well to our task. Another reason why the video classifier does not perform as well as the image classifier is that the video classifiers require large amounts of data to train properly which, due to manual annotation limits, is not available for our dataset.



(a)

(b)

Figure 6.1: Precision and Recall for distracted driving class for (a) Random frame and (b) Single frame for different thresholds.

Training Details We train all our image-based models using Adam optimizer. The best model was trained with learning rate of 0.01, batch size 16, and utilized weight decay for regularization purposes. We train all the models for a maximum of 10 epochs with the total training time of around 12 hours on 4 Nvidia GTX 1080Ti GPU.

For the video classifier models, we use SGD (Stochastic Gradient Descent) with momentum and set the learning rate to 0.1. We use a batch size of 32 for the video classifiers and train both the models for a maximum of 60 epochs each. We also use weight decay as a means of regularization for the model.

Validation and Robustness of Classifier To validate the generalizability of our proposed method, we create a held-out test set from our collected dataset (dataset that was not previously used in any step of training). We randomly sampled $5{,}472$ snaps from our collected dataset ($1,404$ positive, $4,068$ negative). We did not place any geographic/temporal constraints on selecting these posts. On this held-out set, we see that all methods achieve a high accuracy of at least 0.93, as shown in Table 6.4.

In the above section, we show that our proposed deep learning approach that leverages the content of the snap can be used to detect distracted driving content snaps successfully (RQ5).

Table 6.4: Performance of models on held-out set.

Type	Architecture	Accuracy	F1-Score
Image-Based (Majority Voting)	ResNeXt-50	**0.953**	**0.91**
	ResNet-34	0.948	0.894
	WideResNet	0.951	0.904
Video-Based	ResNet-34	0.93	0.859
	ResNeXt-101	**0.942**	**0.859**

6.4 Characterizing Temporal and Spatial Patterns

In this section, we first measure the extent of distracted driving content posting behavior across various cities on the platform. Temporal patterns have proven to be useful for analyzing trends; we perform temporal analysis on our dataset to understand when such type of behavior (posting distracted driving content) is prevalent. Further, we conduct spatial analysis to explore interesting patterns across and within each city to determine if such behavior is concentrated on certain parts of the city or is spread across uniformly.

6.4.1 Extent of distracted driving content

Related to RQ1, we want to understand the extent of posting distracted driving content across various cities. To measure this, we applied our deep-learning classifier built in the previous section on all the snaps (6.43M) we collected. We discovered that around 23.56% of the snaps in our dataset consisted of distracted driving content. Further, we analyzed which cities were exhibiting such behavior the most, and present it in Figure 6.2.

We observe that middle-eastern (Riyadh, Baghdad) and Indian cities (Chandigarh, Amritsar, Ahmedabad) were posting such content in high percentages ($> 35\%$). Such behavior was found to be lower in European and American cities, and we find that there is not even a single European or American city in the list of top-20 cities. Moreover, the first American city (Fremont, CA) that has a high percentage (22.26%) of distracted-driving content has only very few total numbers of snaps (4,042).

Observation 1 (Regional Effect)**.** *The trend of posting distracted driving content on Snapchat is predominantly higher in Middle-Eastern and cities in Indian sub-continent, as compared to other cities across the world.*

6.4.2 Temporal Analysis

We investigate how driving content posting behavior differs across time. In Figure 6.3(a), we present the hour-wise distribution of (i) when users post distracted driving content, (ii) when users post any form of content. We can see that the distracted driving content is approximately a uniform fraction of all the posts across the day. Users are often more active during the night-time (6PM-2AM), posting 73.51% more posts per hour in this period relative to the frequency

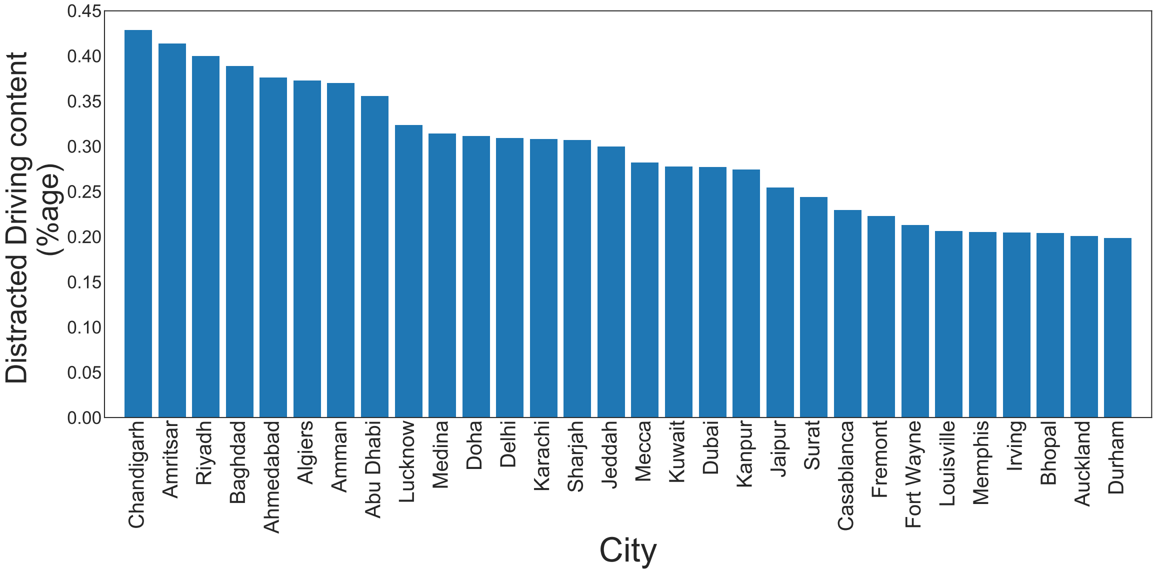


Figure 6.2: The top 30 cities in our dataset ordered based on the ratio of driving snaps to the total snaps.

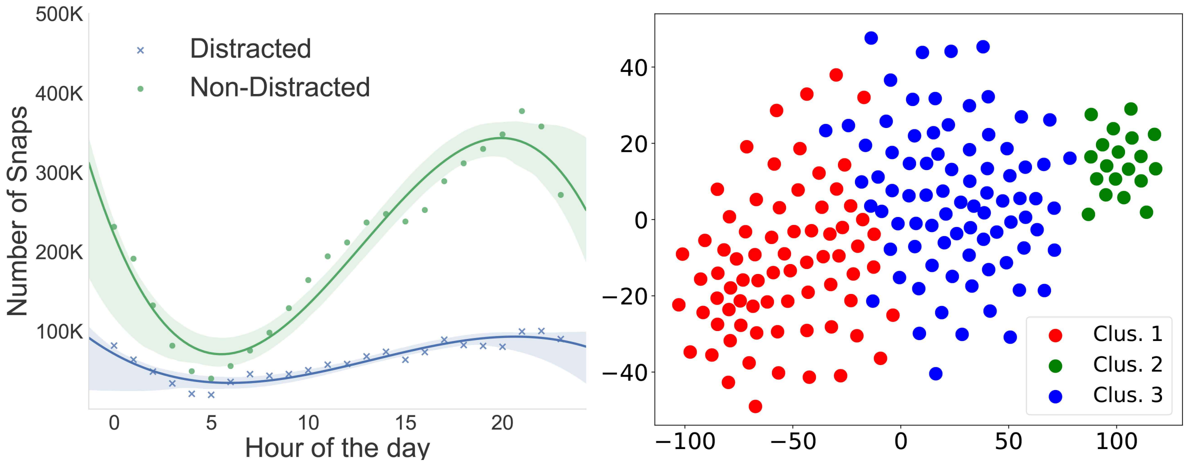
of posting over other hours of the day. We observe a similar trend for driving snaps, where the number of driving snaps posted per hour during the evening to night window is found to be 77.83% more than the rest of the day.

Further, to show that the driving snaps are a uniform fraction of the overall snaps, we compute the correlation between the number of driving snaps posted and the number of total snaps posted in every hour for the entire month of the data collected and find it highly correlated with a Pearson correlation coefficient of 0.9545. We can also observe that a sharp drop in non-distracted-driving content is not complemented with a similar drop in the distracted driving content posting. Due to this, we observe a pattern of higher distracted driving content posting activity through the night, and into the hours of the morning.

Observation 2 (Night-time Driving)**.** *The incidence of posting while driving behavior over the night is more pronounced than other forms of content posting during the same hours.*

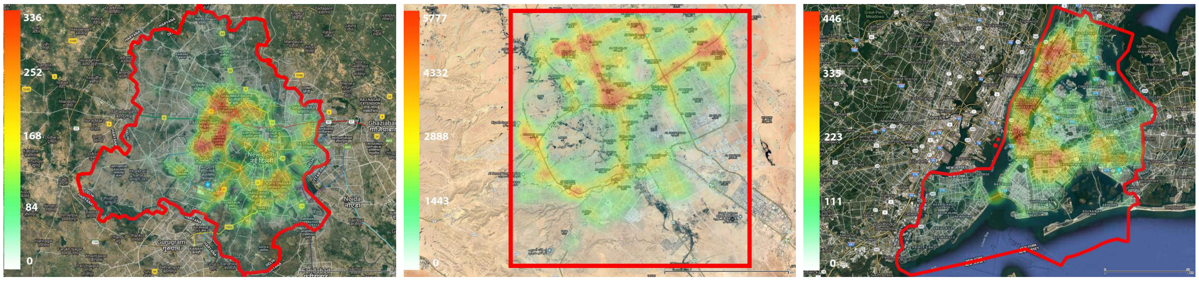
We further investigate the different temporal patterns that exist across different cities. We cluster the fraction of distracted driving snaps posted per hour over the entire week for each city. Using silhouette score coefficient [242] and also Elbow method [279], we estimated the number of clusters to be 3 for K-means clustering. We show the two-component T-SNE representation [186], along with the cluster label for each city to show the efficacy of the clustering in Fig 6.3 (b). From the figure, we can see that the clustering so obtained separates the cities well. In the 3 clusters we obtained, we observed that first cluster corresponded to most European cities (containing 80% of European cities we analyzed). The second cluster consisted only of Indian (7) and Middle-Eastern cities (12). The final cluster consisted of primarily American cities (containing 86% of American cities we analyzed.

Observation 3 (Temporal Clustering)**.** *Temporal patterns exhibited by different cities can be meaningfully clustered, and indicate overall geographical and cultural patterns.*



(a) (b)

Figure 6.3: (a) Diurnal trends (for both the driving and non-driving content classes). The line plots denote the regression fit of the trends. (b) Cities clustered according to their temporal patterns.



(a) (b) (c)

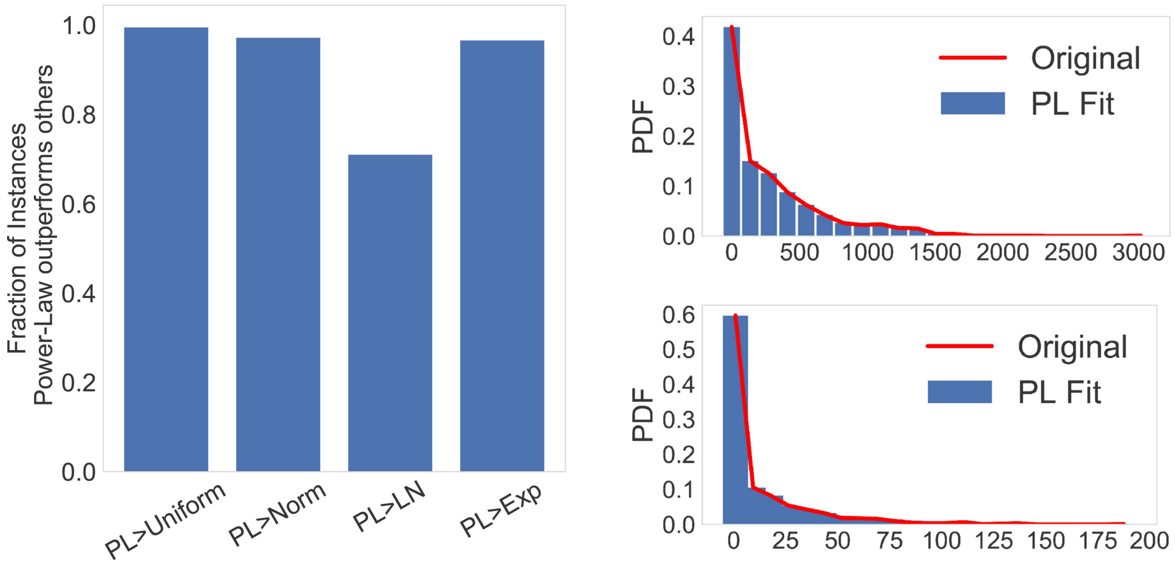
Figure 6.4: Spatial analysis (frequency distribution plots) of three cities (from Table 6.1)

We can observe the presence of temporal patterns in distracted driving content posting behavior, which answers RQ4. This analysis could be used by platform designers or policy makers to target cities at a specific time of the day by discouraging or warning users about this type of behavior.

6.4.3 Spatial Analysis

Previously, spatial analysis on SnapMaps has been used to show that usage of Snapchat, while posting publicly to maps has been concentrated [135]. We use spatial analysis to investigate these insights further while focusing on distracted driving content posting behavior.

In Figure 6.4, we show the spatial distribution of distracted driving content snaps for three popular cities ((a) Delhi, (b) Riyadh and (c) New York City). We can see that for these cities the distribution is concentrated on small regions on the map. To measure if the distracted driving content snaps are concentrated or not, we model the distribution of the number of distracted



(a)

(b)

Figure 6.5: (a) Power-Law distribution fits the best for most of the cities, in comparison to other candidate distributions. (b) Sample fits under Power-Law distribution shown for (top) Riyadh and (bottom) Delhi.

driving content snaps per tile for each city with a known parametric family of distributions. Concentrated distracted driving content snaps will follow a power-law (PL) distribution, as compared to uniform distracted driving content snaps which will follow a uniform distribution. We try to fit multiple distributions (power-law, gaussian, log-normal, and exponential) on all cities per tile to model distracted driving content content distribution. We discover that power-law distribution fits better than all the other candidate distributions for the majority of the cities when compared using log-likelihood and BIC metrics. We plot the percentage of instances for which power-law distribution fits better than other candidate distributions in Figure 6.5(a). We also show power-law distribution fit for two cities - Riyadh (top) and Delhi (bottom) and observe that the fits are visually accurate.

Observation 4 (Concentrated Driving Content)**.** *For most of the cities across the world, the distracted driving content posting behavior is geographically concentrated to only a few tiles and not uniformly distributed across the city.*

Another interesting pattern that we observe was that for certain cities, the distracted driving content was observed to be higher on major roads. For example, in Riyadh's heatmap (Figure 6.4(b)), we can see two major roads having a higher concentration of distracted driving snaps. However, we cannot quantify this pattern across all cities as we do not have access to underlying road and highway data and leave this pattern quantification as future work.

We discovered useful insights about distracted driving content posting behavior within and across cities, thus answering RQ3. Such insights can be used to develop interventions based on geographic areas.

6.5 Characterizing Users

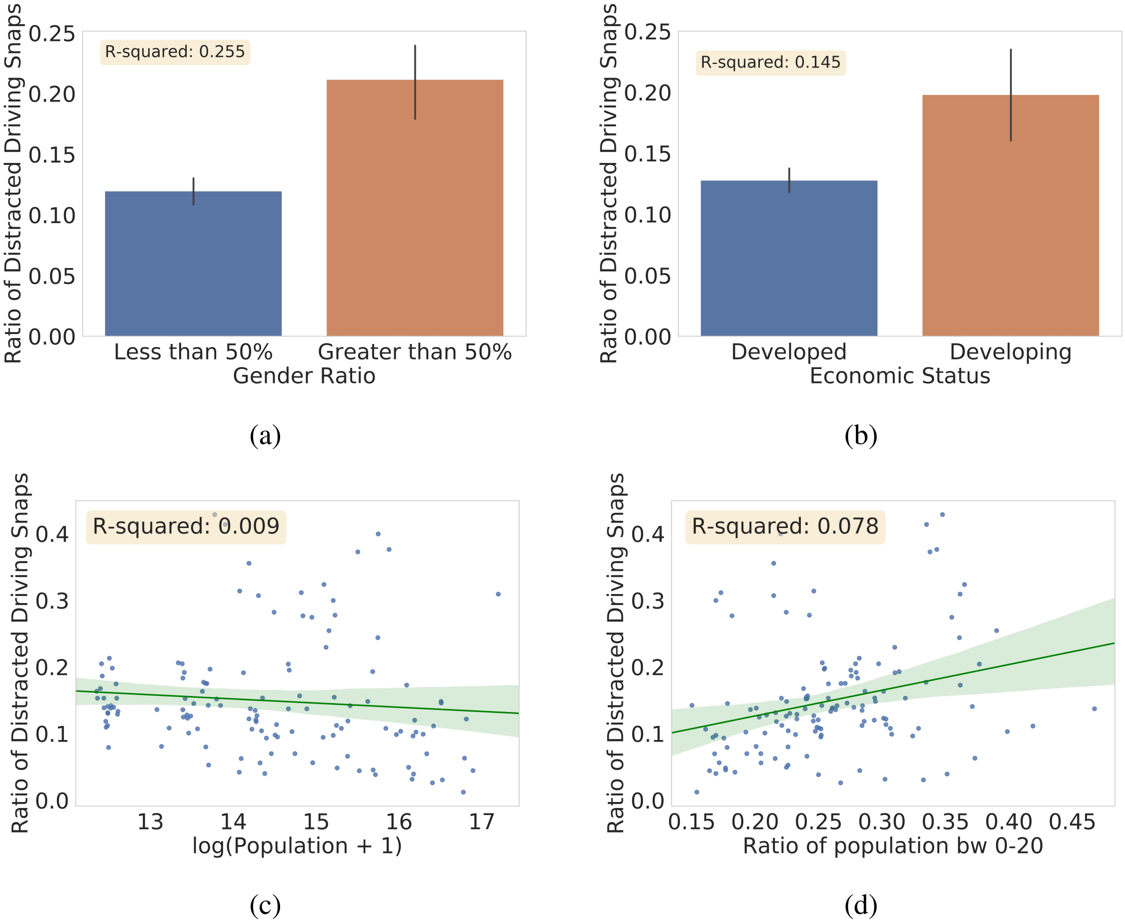
For our investigation into the demographics of the user (RQ2), we aim to understand how the demographics of a particular city affect the number of driving snaps.

6.5.1 Explanatory Variables

Most previous work in risk-taking has focused on two important characteristics of individuals indulging in risk-taking activities [168, 184], namely gender and age. In this work, we extend their work and investigate the role of gender and age in a user's proclivity to create distracted driving content. Therefore, we examine these two features - gender and age distribution for each city. Additionally, since Snapchat is a popular Internet-based platform, it is imperative to understand the economic influences that might affect the type of usage of the platform. Therefore, we use the development status of a country in which the city is as one of the control variables. We classify the countries of the world in our dataset as either *developed* or *developing* based on the definition of developed nations given in CIA's world factbook [8]. The economic status of the city further acts as a proxy for various other additional variables for which data is less readily available such as smartphone penetration, social media usage, and availability of public transportation facilities. We also account for certain control variables such as the total number of snaps posted in the city and the population of the city. We obtain the population estimate for each city from `worldpopulationreview.com`, where we use the latest estimate available. Similarly, we obtain the gender ratio statistic from the latest available census data that has been aggregated on `citypopulation.de`. However, the website does not provide us with the latest data for all the cities. In such cases, we take the latest gender ratio available and assume that it remains constant for the city. For computing the age-distribution, we used the statistics from `citypopulation.de`, and for cities where the data was not available - census data for the respective country was obtained. It is possible that for statistics such as gender and age, the statistics across cities might have been computed for different years. To account for this discrepancy, we use age and gender variables as a percentage over the total population. Finally, we did not include cities for which we did not have satisfactory census data, which left us with 130 cities.

6.5.2 Effect of Variables

We investigate the relationship between the variables mentioned above and the number of distracted driving snaps posted from each city, based on which we observe some interesting patterns. From Figure 6.6(a), we can observe that the distracted driving snaps ratio for cities where the gender ratio is in favor of males is roughly 77% more than that of the cities where the gender ratio is in favor of females ($t = 6.62$, $p < 0.001$). Similarly, from Figure 6.6(b), we observe that distracted driving snaps ratio posted in the developing cities is roughly 55% more than that of the developed cities ($t = 4.66$, $p < 0.001$). In Figure 6.6(c), we present the scatter plot of the population of a city (log scale) with the number of driving snaps posted. We can see that there is a small negative slope, possibly implying that cities with the larger population have a lower number of driving snaps. Interestingly, we note that the slope in the case of the ratio of the population below $0 - 20$ is positive ($R^2 = 0.078$, $p < 0.01$), suggesting that cities with a higher ratio of population in the age group of $0 - 20$ have a higher number of driving snaps.



(a)

(b)

(c)

(d)

Figure 6.6: Scatter plot of how number of driving snaps is affected by different variables: (a) Gender Ratio: Ratio of Males to Females (b) Development status of the city (c) Population of the city, (d) Ratio of population between ages 0 and 20

6.5.3 Statistical Model

We are interested in explaining the number of distracted driving snaps posted from every city. We assume a linear relationship between the number of distracted driving snaps and the other variables discussed previously. We transform all the count variables to log-scale to stabilize their variances. The explaining variables (or independent variables) along with the dependent variable we use to model are shown in Table 6.5. Besides the explaining variables - we also use the number of total snaps as a natural control for the popularity of Snapchat in the city. We present the results of the regression on all the 130 cities for which we were able to get satisfactory data in Table 6.6.

Table 6.5: List of dependent variables used to estimate the number of driving snaps posted.

Variable Name	Description	Min.	Max.
Independent Variables			
$\log(Pop. + 1)$	Population	12.35	17.19
$Age < 20$	% of pop. <20	15.0	46.7
$20 < Age < 40$	20 >% of pop. <40	19.4	58.3
$40 < Age$	% of pop. >40	14.1	60.5
Male ratio	Ratio of Male pop.	0.458	0.756
$\log(TS + 1)$	# Total Snaps	5.412	13.813
Dependent Variable			
$\log(DS + 1)$	# Driving Snaps	2.08	12.89

Analyzing the results, we can see that the term Total Snaps (TS) introduced as a control variable behaves as expected. The effect of the variable is significant and positively related, with a one percent rise in the log number of snaps posted associated with a 1.21% rise in the log number of distracted driving snaps. We can also see that the population of a city has a significant negative effect. This could perhaps be explained by the fact that as the cities grow in population, the traffic and congestion on the road also increases, leading to more time spent on paying attention to the road as compared to that spent on a phone.

Connecting back to our RQ2, we want to figure out what demographics of users are more inclined to indulge in distracted driving content posting behavior. We first investigate the role of gender and its contribution to the number of distracted driving snaps across cities. It has often

Table 6.6: Regression models for number of distracted driving snaps (N=130).

	Dependent variable	
	$\log(DS + 1)$	
	Coeffs(Err.)	LR ChiSq
Intercept	$-6.86(1.48)^{***}$	
$Males$	$0.05(0.01)^{***}$	607.14^{***}
$Age < 20$	$5.85(1.46)^{***}$	33.72^{***}
$20 < Age < 40$	$1.92(1.56)$	0.60
$Age > 40$	$2.38(1.40)^{\cdot}$	$2.89^{\cdot}$
$Developing$	$0.19(0.12)^{\cdot}$	81.48^{***}
$\log(Pop. + 1)$	$-0.21(0.03)^{***}$	51.53^{***}
$\log(TS + 1)$	$1.21(0.03)^{***}$	2269.99^{***}
R^2 coefficient	0.9593	

Note:$^{***}p < 0.001$, $^{**}p < 0.01$, $^{\cdot}p < 0.1$

been shown that proclivity of taking risk is higher among males [168, 184]. We verify the same hypothesis in our regression model, where we observe that the percentage of the male population has a significant, positive, and large effect. A one percent increase in the male ratio would lead to a 0.05% rise in the log number of distracted driving snaps.

Observation 5 (Role of Gender)**.** *Cities with higher male ratio are more likely to produce more distracted driving snaps.*

Another popular result of the edgework framework is that younger people are more likely to participate and indulge in risk-taking activities. In our model, we introduced 3 variables as percentage of individuals less than 20 years of age ($Age < 20$), between 20 and 40 years of age ($20 < Age < 40$), and above 40 ($Age > 40$). We discovered that $20 < Age$ has a significant positive effect on the number of distracted driving snaps posted in the city. However, the other two variables did not have any significant effect. Though this result is significant, it is also probably biased as Snapchat is a platform that is primarily used by young people; hence, there is a possibility that this observation just might be capturing that effect.

Observation 6 (Role of Age)**.** *Cities with higher proportion of young people are more likely to post distracted driving snaps than cities with higher proportion of older people.*

Additionally, we see that there is an effect of whether the city is developed or developing ($Developing$) on the number of distracted driving snaps that get posted. We discover that if a city is in a developing nation, then there are higher chances of distracted driving snap posting behavior. This is in accordance with the overall spatial and temporal pattern observed, the cities being ranked consistently higher in distracted driving snap posting behavior were mostly cities from developing countries.

Observation 7 (Effect of Development)**.** *Users from cities in developing world are more likely to post distracted driving snaps.*

6.6 Discussion

6.6.1 Research Questions

RQ1 relates to the extent of distracted driving snaps are posted on Snapchat across cities. The question tries to estimate the prevalence of such type of risk-taking behavior on social media platforms, thus quantifying the importance of studying such problems. We discovered that distracted driving snaps form 23.56% of total snaps posted across 173 cities. Further, we also noticed that such behavior is more prevalent in Middle-Eastern and sub-continent Indian cities(accounting for 72.4% of distracted driving snaps overall). By answering **RQ**3, we investigated the spatial patterns of distracted driving content posting behavior. We discovered that such content is posted in certain regions of the city; and is not uniform across the city, thus, showing that distracted driving content posting behavior is concentrated. However, we were unable to analyze these hotspots for the underlying demographic and geographical features to understand the reason behind such concentration - largely due to the lack of data at that granularity. **RQ**4 is focused on determining temporal patterns behind distracted driving content posting behavior. We made key observations based on temporal analysis of the behavior across cities. We discovered that most of such content is posted heavily during night-time. Further, we were also able to discover strong regional effects - where the clusters formed on clustering the fraction of snaps posted each hour of the week segmented into clusters comprising majorly of European, American and Mid-Eastern cities.

One of the key frameworks proposed by sociologists to explain risk-taking literature has been *edgework*. The framework, besides defining voluntary risk-taking behavior and applying it to different settings, also proposed characteristics that define the users who are inclined to take such risks. The observations made about such voluntary risk-takers was based on the concept of an illusory sense of control, where a user feels that they have more control of the situation than they actually do. The theory discovered that males and young people generally felt more of such an illusory sense of control. We tested whether the theories put forward by the *edgework* framework also hold for the case of distracted driving content posting behavior on social media platforms. We attempted to answer this in **RQ**2. We discovered, in concurrence with the theory, that males are more inclined to participate in such voluntary risk-taking behavior. Further, we also discovered that younger people are more inclined to exhibit such behavior, another key characteristic proposed by the framework. Another key point put forth by the theory was that individuals who were of a social system that exhibited much larger control over their life ended up participating in such behavior in seek of a high-stakes feeling of control over the situation. We hypothesized that this could relate to the economic situation of a particular city - and tested if individuals from developing regions (instead of developed) were more likely to participate in risks or not. We discovered that we do see the effect of the economic status of the city. However, we only treat economic status as a proxy for control; many other factors such as political and cultural could be considered, which are hard to obtain and quantify.

Finally, to be able to answer any of the **RQ**s as mentioned earlier, we needed to figure out how can we detect if a particular snap is an example of distracted driving content or not. Due to the large scale of our study, it is infeasible to label the entire dataset manually. Hence, we answered **RQ**5 by proposing a deep learning classifier and were able to achieve high precision and recall. Further, we even tested the robustness of the trained classifier to show that the proposed method performs robustly on an held-out set.

6.6.2 Implications

Our paper provides a robust way of detecting if the content posted on Snapchat is an instance of distracted driving content or not. Further, our results provide insights into the extent of such behavior on a popular social media platform Snapchat, and spatial, temporal and demographics related patterns. We believe that the platform owners and policymakers can leverage insights put forward by our work to develop educational campaigns and interventions. We discuss some of the suggestions below:

Location-Based: One of our key insights (Insight 1) was that distracted driving content posting behavior is prevalent mostly in Middle Eastern and Indian cities. Thus, some of the educational campaigns could be focused only on these regions and can be disseminated within the platform itself. Another insight that could be crucial in designing platform-based interventions is that such behavior is concentrated only in certain regions of cities. Combined with the proposed deep learning classifier, such content from these hotspots can be analyzed if they are instances of distracted driving or not, and not shown to the public. In the case of Snapchat specifically, users post such content on SnapMaps to gain popularity from the general public; however, if such content is not allowed to be posted on the platform from these regions, there is a possibility that it might discourage the individuals from creating such content. However, this requires more experimentation to determine if such a form of intervention can be useful or not.

Time-Based: Our work made a useful insight about nighttime driving, indicating that such content is generally posted late in the night (Insight 2). This insight could be leveraged to issue educational notifications at that time of the day when such at-risk users could be active.

Demographics-Based: The major insights we draw from our regression analysis was the role of age and gender in characterizing the users who participate in such behavior. We discovered through insights 5 and 6 that young individuals and males are more likely to participate in such behavior. If a platform has a way of inferring identities of their users, it could be leveraged in combination with the other insights to create targeted interventions and educational campaigns for these specific demographics.

We are aware that a social media platform has other constraints while issuing notifications, such as the number of them, and restricting users not to share certain types of content, which could potentially lead to violation of their freedom to express. However, a lot of the interventions mentioned above methods and educational campaigns can be combined with the proposed deep learning-based detection approach to give the platforms, access to multiple intervention designs, thus providing flexibility to the platform. Since such interventions can also act in unintended ways (such as suggesting risk-takers not to perform risk-taking behavior; hence, actually motivating them), more analysis needs to be done before proceeding forward.

6.6.3 Threats to Validity

Like any quantitative study, our work is subject to threats to validity. We try to enumerate biases, issues, and threats to the validity of our study by following a framework for inferring biases and pitfalls while analyzing social data by Olteanu et al.[222]. First, our work is based on the data collected on Snapchat, mostly through SnapMap. A key data issue is that of representativeness - our collected data, though might not be geographically or temporally biased (since we collected data across the world and for a large amount of time), it can still be that we are collecting data disproportionately from regions that post more frequently publicly on SnapMaps rather than Snapchat in general. Another representative issue is that we are linking Snapchat usage data with that of census data in general; where Snapchat users might not be representative of the entire cities population. We try to discount this representation bias by including appropriate control variables, but still, some of the bias might exist in our analysis. Additionally, our dataset might also contain temporal bias as during our one-month long data collection; it might be possible that some cities might be observing festival-related holidays or some events. This might have introduced a disproportion in the number of snaps collected from each city. A significant source of data bias in our analysis is the use of census data. Firstly, we were not able to obtain data for each city and thus had to omit certain cities from our analysis. Secondly, census data is obtained from different years, and finally, the census data for different cities are taken from different sources.

For the annotation required for training deep learning classifier, we used a limited number of annotators, which might result in subjective interpretation. We attempted to mitigate this threat by using majority voting and computing inter-annotator agreement rate. Finally, our statistical modeling required multiple parameters that were related to the operationalization of theories that exist in literature. Some of these parameters might not be capturing the factors that we intended to capture or that the theories captured. It could be possible that our analysis might be applicable only for Snapchat and might not generalize well for other platforms and also for other risk-taking behavior.

6.7 Related Work

Besides the relevant theories and framing discussed in “Development of Research Questions”, there are other related work that should be discussed. We discuss them here:
Recently, there have been some studies on analyzing risk-taking behavior on social media for different voluntary activities. Lamba et al. covered a much broader case of dangerous selfies, where users posted a perilous self-portrait in dangerous situations such as at an elevation, with a firearm, or inside a water body[161]. They also showed that users often engage in risk-taking activities while taking selfies to post on social media. Of the 232 deaths due to taking dangerous selfies, 12 could be attributed to driving-related incidents. The authors presented deep-learning models to distinguish between potentially dangerous and non-dangerous selfies [215]. Similarly, Hart examined young individuals’ participation in posting nude self-portraits on Tumblr [111]. There has been a normative increase in individuals dabbling in risk-taking behavior as a result of various other social media trends such as the Tide Pod Challenge [210], the Cinnamon Challenge [93], the Salt and Ice Challenge [243] and the Fire Challenge [10, 17]. However our work is the first in analyzing the specific behavior of distracted driving content posting on social media. Further we extend the popular voluntary risk-taking *edgework* framework to social media platforms.

6.8 Conclusions

In this work, we investigate the widespread prevalence of distracted driving content posting behavior. We specifically focus on a popular social media platform, Snapchat, and by analyzing the publicly posted stories, we characterized the extent of distracted driving content that exists on such platforms.

Our first contribution is proposing a deep learning based classifier to detect if a content posted is distracted driving or not. Grounding our work in risk-taking literature, we aim to test out the theories put forth by sociologists in terms of risk-taking behavior in the offline world in the context of distracted driving content posting behavior on social media platforms and test them. To this end, we proposed and answered multiple RQs related to extent, spatial, temporal and demographic patterns of such behavior across 173 cities.

We made the following key observations related to the few RQs - the demographics such as age and gender play a key role in the proclivity to post distracted driving content. Further, we also discovered that there exists spatial and temporal patterns in distracted driving content behavior posting across cities. We hypothesize that the insights derived from this study can be used to design targeted intervention and educational campaigns to curb such risk-taking behavior.

Privacy and Ethics: We collect data from SnapMaps, a geographical interface for Snapchat, which is publicly available. The data posted on the platform is already anonymized, and we neither collect nor use any personally identifiable information for our analysis. For variables extracted from the census, we only use the variables as is collected by the respective country's census department.

Subhabrata Mukherjee, Hemank Lamba, Gerhard Weikum. ”Experience-aware item recommendation in evolving review communities”. 2015 IEEE International Conference on Data Mining (ICDM), 2015.

CHAPTER 7

MODELING EXPERIENCE IN RECOMMENDATION SYSTEMS

Current recommender systems exploit user and item similarities by collaborative filtering. Some advanced methods also consider the temporal evolution of item ratings as a global background process. However, all prior methods disregard the *individual evolution* of a user's *experience* level and how this is expressed in the user's *writing* in a review community. In this paper, we model the *joint evolution* of *user experience*, interest in specific *item facets*, *writing style*, and *rating behavior*. This way we can generate individual recommendations that take into account the user's maturity level (e.g., recommending art movies rather than blockbusters for a cinematography expert). As only item ratings and review texts are observables, we capture the user's experience and interests in a *latent model* learned from her reviews, vocabulary and writing style. We develop a generative HMM-LDA model to trace user evolution, where the Hidden Markov Model (HMM) traces her latent experience progressing over time — with solely user reviews and ratings as observables over *time*. The facets of a user's interest are drawn from a Latent Dirichlet Allocation (LDA) model derived from her reviews, as a function of her (again latent) experience level. In experiments with five real-world datasets, we show that our model improves the rating prediction over state-of-the-art baselines, by a substantial margin. We also show, in a use-case study, that our model performs well in the assessment of user experience levels.

Collaborative filtering algorithms are at the heart of recommender systems for items like movies, cameras, restaurants and beer. Most of these methods exploit user-user and item-item similarities in addition to the history of user-item ratings — similarities being based on latent factor models over user and item features [149], and more recently on explicit links and interactions among users [101][296].

All these data evolve over *time* leading to bursts in item popularity and other phenomena like anomalies[103]. State-of-the-art recommender systems capture these temporal aspects by introducing global bias components that reflect the evolution of the user and community as a whole[148]. A few models also consider changes in the social neighborhood of users[185]. What is missing in all these approaches, though, is the awareness of how *experience* and *maturity* levels evolve in *individual users*.

Individual experience is crucial in how users appreciate items, and thus react to recommendations. For example, a mature cinematographer would appreciate tips on art movies much more than recommendations for new blockbusters. Also, the facets of an item that a user focuses on change with experience. For example, a mature user pays more attention to narrative, light effects, and style rather than actors or special effects. Similar observations hold for ratings of wine, beer, food, etc.

Our approach advances state-of-the-art by tapping review texts, modeling their properties as latent factors, using them to explain and predict item ratings as a function of a user's experience evolving over time. Prior works considering review texts (e.g., [155, 193, 208, 290, 293]) did this only to learn topic similarities in a static, snapshot-oriented manner, without considering time at all. The only prior work [194], considering time, ignores the text of user-contributed reviews in harnessing their experience. However, user experience and their interest in specific item facets at different timepoints can often be observed only *indirectly* through their ratings, and more *vividly* through her vocabulary and writing style in reviews.

Use-cases: Consider the reviews and ratings by two users on a "Canon DSLR" camera about the facet camera *lens*.

- *User 1: My first DSLR. Excellent camera, takes great pictures in HD, without a doubt it brings honor to its name. [Rating: 5]*
- *User 2: The EF 75-300 mm lens is only good to be used outside. The 2.2X HD lens can only be used for specific items; filters are useless if ISO, AP,... are correct. The short 18-55mm lens is cheap and should have a hood to keep light off lens. [Rating: 3]*

The second user is clearly more experienced than the first one, and more reserved about the lens quality of that camera model. Future recommendations for the second user should take into consideration the user's maturity. As a second use-case, consider the following reviews of Christopher Nolan movies where the facet of interest is the non-linear *narrative style*.

- *User 1 on Memento (2001): "Backwards told is thriller noir-art empty ultimately but compelling and intriguing this."*
- *User 2 on The Dark Knight (2008): "Memento was very complicated. The Dark Knight was flawless. Heath Ledger rocks!"*
- *User 3 on Inception (2010): "Inception is a triumph of style over substance. It is complex only in a structural way, not in terms of plot. It doesn't unravel in the way Memento does."*

The first user does not appreciate complex narratives, making fun of it by writing her review backwards. The second user prefers simpler blockbusters. The third user seems to appreciate the complex narration style of Inception and, more of, Memento. We would consider this maturity level of the more experienced User 3 to generate future recommendations to her.

Approach: We model the joint evolution of *user experience*, interests in specific *item facets*, *writing style*, and *rating behavior* in a community. As only item ratings and review texts are directly observed, we capture a user's experience and interests by a latent model learned from her reviews, and vocabulary. All this is conditioned on *time*, considering the *maturing rate* of a user. Intuitively, a user gains experience not only by writing many reviews, but she also needs

Table 7.1: Vocabulary at different experience levels.

Experience	Beer	Movies	News
Level 1	bad, shit	stupid, bizarre	bad, stupid
Level 2	sweet, bitter	storyline, epic	biased, unfair
Level 3	caramel finish, coffee roasted	realism, visceral, nostalgic	opinionated, fallacy, rhetoric

to continuously improve the quality of her reviews. This varies for different users, as some enter the community being experienced. This allows us to generate individual recommendations that take into account the user's maturity level and interest in specific facets of items, at different timepoints.

We develop a generative HMM-LDA model for a user's evolution, where the Hidden Markov Model (HMM) traces her latent experience progressing over time, and the Latent Dirichlet Allocation (LDA) model captures her interests in specific item facets as a function of her (again, latent) experience level. The only explicit input to our model is the ratings and review texts upto a certain timepoint; everything else – especially the user's experience level – is a latent variable. The output is the predicted ratings for the user's reviews following the given timepoint. In addition, we can derive interpretations of a user's experience and interests by salient words in the distributional vectors for latent dimensions. Although it is unsurprising to see users writing sophisticated words with more experience, we observe something more interesting. For instance in specialized communities like `beeradvocate.com` and `ratebeer.com`, experienced users write more descriptive and *fruity* words to depict the beer taste (cf. Table 7.5). Table 7.1 shows a snapshot of the words used at different experience levels to depict the facets *beer taste*, *movie plot* and *bad journalism*, respectively.

We apply our model to 12.7 million ratings from 0.9 million users on 0.5 million items in five different communities on movies, food, beer, and news media, achieving an improvement of 5% to 35% for the mean squared error for rating predictions over several competitive baselines. We also show that users at the same (latent) experience level do indeed exhibit similar vocabulary, and facet interests. Finally, a use-case study in a news community to identify experienced *citizen journalists* demonstrates that our model captures user maturity fairly well.

Contributions: To summarize, this paper introduces the following novel elements:

a) This is the first work that considers the progression of user experience as expressed through the text of item reviews, thereby elegantly combining text and time.

b) An approach to capture the natural *smooth* temporal progression in user experience factoring in the *maturing rate* of the user, as expressed through her writing.

c) Offers interpretability by learning the vocabulary usage of users at different levels of experience.

d) A large-scale experimental study in *five* real world datasets from different communities like movies, beer and food; and an interesting use-case study in a news community.

7.1 Overview

7.1.1 Model Dimensions

Our approach is based on the intuition that there is a strong coupling between the *facet preferences* of a user, her *experience*, *writing style* in reviews, and *rating behavior*. All of these factors jointly evolve with *time* for a given user.

We model the user experience progression through discrete stages, so a state-transition model is natural. Once this decision is made, a Markovian model is the simplest, and thus natural choice. This is because the experience level of a user at the current instant t depends on her experience level at the previous instant t-1. As experience levels are latent (not directly observable), a Hidden Markov Model is appropriate. Experience progression of a user depends on the following factors:

- *Maturing rate* of the user which is modeled by her *activity* in the community. The more engaged a user is in the community, the higher are the chances that she gains experience and advances in writing sophisticated reviews, and develops taste to appreciate specific facets.
- *Facet preferences* of the user in terms of focusing on particular facets of an item (e.g., narrative structure rather than special effects). With increasing maturity, the taste for particular facets becomes more refined.
- *Writing style* of the user, as expressed by the language model at her current level of experience. More sophisticated vocabulary and writing style indicates higher probability of progressing to a more mature level.
- *Time difference* between writing successive reviews. It is unlikely for the user's experience level to change from that of her last review in a short time span (within a few hours or days).
- *Experience level difference*: Since it is unlikely for a user to directly progress to say level 3 from level 1 without passing through level 2, the model at each instant decides whether the user should stay at current level l, or progress to l+1.

In order to learn the *facet preferences* and *language model* of a user at different levels of experience, we use *Latent Dirichlet Allocation* (LDA). In this work, we assume each review to refer to exactly one item. Therefore, the facet distribution of items is expressed in the facet distribution of the review documents.

We make the following assumptions for the generative process of writing a review by a user at time t at experience level e_t:

- A user has a distribution over *facets*, where the facet preferences of the user depend on her experience level e_t.
- A facet has a distribution over *words* where the words used to describe a facet depend on the user's vocabulary at experience level e_t. Table 7.2 shows salient words for two facets of Amazon movie reviews at different levels of user experience, automatically extracted by our latent model. The facets are latent, but we can interpret them as *plot/script* and *narrative style*, respectively.

As a sanity check for our assumption of the coupling between user *experience*, *rating behavior*, *language* and *facet preferences*, we perform experimental studies reported next.

Table 7.2: Salient words for two facets at five experience levels in movie reviews.

Level 1: stupid people supposed wouldnt pass bizarre totally cant
Level 2:storyline acting time problems evil great times didnt money ended simply falls pretty
Level 3: movie plot good young epic rock tale believable acting
Level 4: script direction years amount fast primary attractive sense talent multiple demonstrates establish
Level 5: realism moments filmmaker visual perfect memorable recommended genius finish details defined talented visceral nostalgia

Level 1: film will happy people back supposed good wouldnt cant
Level 2: storyline believable acting time stay laugh entire start funny
Level 3 & 4: narrative cinema resemblance masterpiece crude undeniable admirable renowned seventies unpleasant myth nostalgic
Level 5: incisive delirious personages erudite affective dramatis nucleus cinematographic transcendence unerring peerless fevered

7.1.2 Hypotheses and Initial Studies

Hypothesis 1: Writing Style Depends on Experience Level.

We expect users at different experience levels to have divergent Language Models (LM's) — with experienced users having a more sophisticated writing style and vocabulary than amateurs. To test this hypothesis, we performed initial studies over two popular communities[1]: 1) BeerAdvocate (`beeradvocate.com`) with 1.5 million reviews from $33,000$ users and 2) Amazon movie reviews (`amazon.com`) with 8 million reviews from $760,000$ users. Both of these span a period of about 10 years.

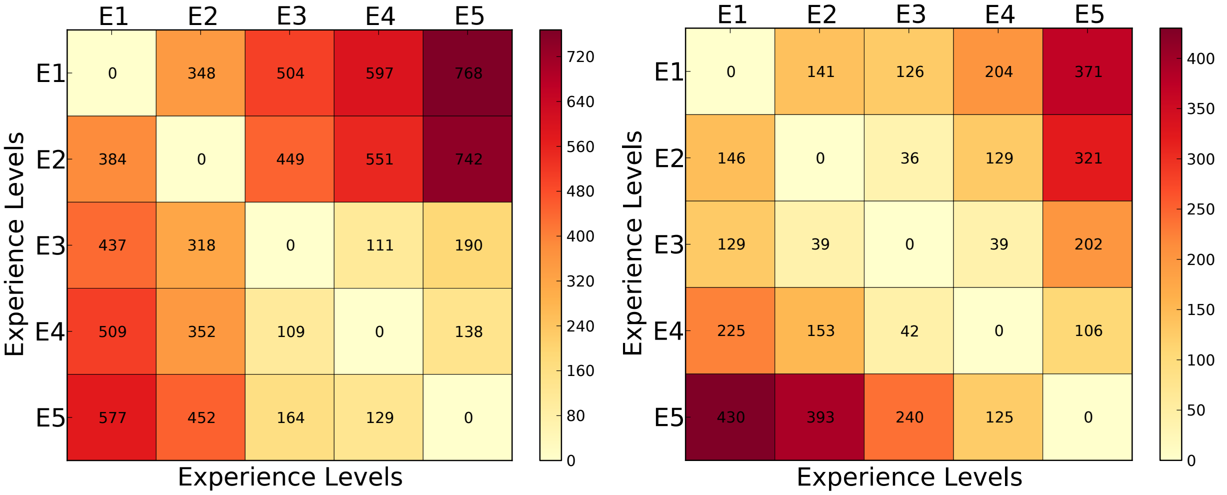
In BeerAdvocate, a user gets *points* on the basis of likes received for her reviews, ratings from other users, number of posts written, diversity and number of beers rated, time in the community, etc. We use this points measure as a proxy for the user's *experience*. In Amazon, reviews get *helpfulness* votes from other users. For each user, we aggregate these votes over all her reviews and take this as a proxy for her experience.

We partition the users into 5 bins, based on the points / helpfulness votes received, each representing one of the experience levels. For each bin, we aggregate the review texts of all users in that bin and construct a unigram language model. The heatmap of Figure 7.1a shows the *Kullback-Leibler* (KL) divergence between the LM's of different experience levels, for the BeerAdvocate case. The Amazon reviews lead to a very similar heatmap, which is omitted here. The main observation is that the KL divergence is higher — the larger the difference is between the experience levels of two users. This confirms our hypothesis about the coupling of experience and user language.

Hypothesis 2: Facet Preferences Depend on Experience Level.

The second hypothesis underlying our work is that users at similar levels of experience have similar facet preferences. In contrast to the LM's where words are *observed*, facets are *latent* so that validating or falsifying the second hypothesis is not straightforward. We performed a

[1] Data available at http://snap.stanford.edu/data/



(a) Divergence of language model as a function of experience.

(b) Divergence of facet preference as a function of experience.

Figure 7.1: KL Divergence as a function of experience.

three-step study:

- We use Latent Dirichlet Allocation (LDA) [28] to compute a latent facet distribution $\langle f_k \rangle$ of *each review*.
- We run Support Vector Regression (SVR) [64] for *each user*. The user's item rating in a review is the response variable, with the facet proportions in the review given by LDA as features. The regression weight $w_k^{u_e}$ is then interpreted as the preference of user u_e for facet f_k.
- Finally, we aggregate these facet preferences for each experience level e to get the corresponding facet preference distribution given by $< \frac{\sum_{u_e} exp(w_k^{u_e})}{\#u_e} >$.

Figure 7.1b shows the KL divergence between the facet preferences of users at different experience levels in BeerAdvocate. We see that the divergence clearly increases with the difference in user experience levels; this confirms the hypothesis. The heatmap for Amazon is similar and omitted.

Note that Figure 7.1 shows how a *change* in the experience level can be detected. This is not meant to predict the experience level, which is done by the model in Section 7.3.

7.2 Building Blocks of our Model

Our model, presented in the next section, builds on and compares itself against various baseline models as follows.

7.2.1 Latent-Factor Recommendation

According to the standard latent factor model (LFM) [147], the rating assigned by a user u to an item i is given by:

$$rec(u,i) = \beta_g + \beta_u + \beta_i + \langle \alpha_u, \phi_i \rangle \tag{7.1}$$

where $\langle ., . \rangle$ denotes a scalar product. β_g is the average rating of all items by all users. β_u is the offset of the average rating given by user u from the global rating. Likewise β_i is the rating bias for item i. α_u and ϕ_i are the latent factors associated with user u and item i, respectively. These latent factors are learned using gradient descent by minimizing the mean squared error (MSE) between observed ratings $r(u,i)$ and predicted ratings $rec(u,i)$: $MSE = \frac{1}{|U|}\sum_{u,i \in U}(r(u,i) - rec(u,i))^2$

7.2.2 Experience-based Latent-Factor Recommendation

The most relevant baseline for our work is the "user at learned rate" model of [194], which exploits that users at the same experience level have similar rating behavior even if their ratings are temporarily far apart. Experience of each user u for item i is modeled as a latent variable $e_{u,i} \in \{1...E\}$. Different recommenders are learned for different experience levels. Therefore Equation 7.1 is parameterized as:

$$rec_{e_{u,i}}(u,i) = \beta_g(e_{u,i}) + \beta_u(e_{u,i}) + \beta_i(e_{u,i}) + \langle \alpha_u(e_{u,i}), \phi_i(e_{u,i}) \rangle \tag{7.2}$$

The parameters are learned using Limited Memory BFGS with the additional constraint that experience levels should be non-decreasing over the reviews written by a user over time.

However, this is significantly different from our approach. All of these models work on the basis of only user *rating behavior*, and ignore the review texts completely. Additionally, the *smoothness* in the evolution of parameters between experience levels is enforced via L_2 regularization, and does not model the *natural* user maturing rate (via HMM) as in our model. Also note that in the above parametrization, an experience level is estimated for each user-item pair. However, it is rare that a user reviews the same item multiple times. In our approach, we instead trace the evolution of users, and not user-item pairs.

7.2.3 User-Facet Model

In order to find the facets of interest to a user, [241] extends Latent Dirichlet Allocation (LDA) to include authorship information. Each document d is considered to have a distribution over authors. We consider the special case where each document has exactly one author u associated with a Multinomial distribution θ_u over facets Z with a symmetric Dirichlet prior α. The facets have a Multinomial distribution ϕ_z over words W drawn from a vocabulary V with a symmetric Dirichlet prior β. Exact inference is not possible due to the intractable coupling between Θ and Φ. Two ways for approximate inference are MCMC techniques like Collapsed Gibbs Sampling and Variational Inference. The latter is typically much more complex and computationally expensive. In our work, we thus use sampling.

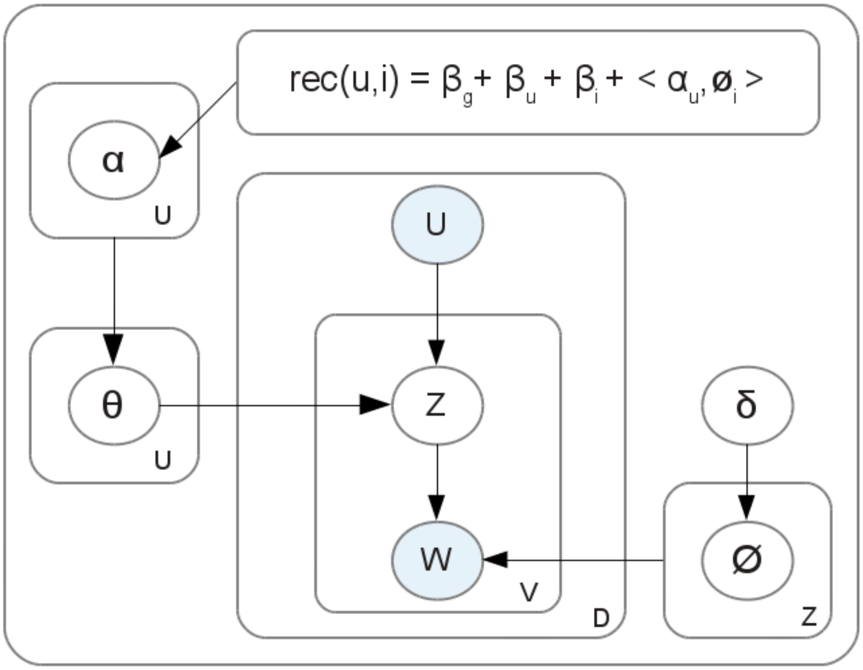


Figure 7.2: Supervised model for user facets and ratings.

7.2.4 Supervised User-Facet Model

The generative process described above is unsupervised and does not take the ratings in reviews into account. Supervision is difficult to build into MCMC sampling where ratings are continuous values, as in communities like `newstrust.net`. For discrete ratings, a review-specific Multinomial rating distribution $\pi_{d,r}$ can be learned as in [175, 236]. Discretizing the continuous ratings into buckets bypasses the problem to some extent, but results in loss of information. Other approaches [155, 193, 208] overcome this problem by learning the feature weights separately from the user-facet model.

An elegant approach using Multinomial-Dirichlet Regression is proposed in [199] to incorporate arbitrary types of observed continuous or categorical features. Each facet z is associated with a vector λ_z whose dimension equals the number of features. Assuming x_d is the feature vector for document d, the Dirichlet hyper-parameter α for the document-facet Multinomial distribution Θ is parametrized as $\alpha_{d,z} = exp(x_d^T \lambda_z)$. The model is trained using stochastic *EM* which alternates between 1) sampling facet assignments from the posterior distribution conditioned on words and features, and 2) optimizing λ given the facet assignments using L-BFGS. Our approach, explained in the next section, follows a similar approach to couple the User-Facet Model and the Latent-Factor Recommendation Model (depicted in Figure 7.2).

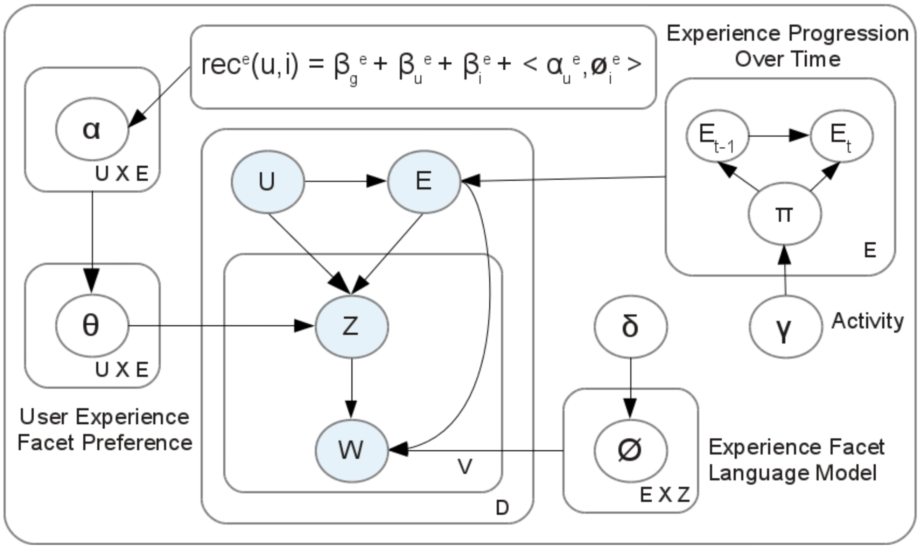

rec^e(u,i) = β_g^e + β_u^e + β_i^e + < α_u^e,ø_i^e >
Experience Progression Over Time
α
U X E
U
E
E_t-1
E_t
π
E
θ
U X E
Z
δ
γ
Activity
User Experience Facet Preference
W
V
D
ø
E X Z
Experience Facet Language Model

Figure 7.3: Supervised model for user experience, facets, and ratings.

7.3 Joint Model: User Experience, Facet Preference, Writing Style

We start with a *User-Facet Model (UFM)* (aka. Author-Topic Model [241]) based on *Latent Dirichlet Allocation* (LDA), where users have a distribution over facets and facets have a distribution over words. This is to determine the facets of interest to a user. These facet preferences can be interpreted as latent item factors in the traditional *Latent-Factor Recommendation Model* (LFM) [147]. However, the LFM is supervised as opposed to the UFM. It is not obvious how to incorporate supervision into the UFM to predict ratings. The user-provided ratings of items can take continuous values (in some review communities), so we cannot incorporate them into a UFM with a Multinomial distribution of ratings. We propose an *Expectation-Maximization (EM)* approach to incorporate supervision, where the latent facets are estimated in an *E-Step* using *Gibbs Sampling*, and *Support Vector Regression* (SVR) [64] is used in the *M-Step* to learn the feature weights and predict ratings. Subsequently, we incorporate a layer for *experience* in the UFM-LFM model, where the experience levels are drawn from a *Hidden Markov Model* (HMM) in the *E-Step*. The experience level transitions depend on the evolution of the user's *maturing rate*, *facet preferences*, and *writing style* over *time*. The entire process is a supervised generative process of generating a review based on the experience level of a user hinged on our HMM-LDA model.

7.3.1 Generative Process for a Review

Consider a corpus with a set D of review documents denoted by $\{d_1 \dots d_D\}$. For *each user*, all her documents are ordered by timestamps t when she wrote them, such that $t_{d_i} < t_{d_j}$ for $i < j$. Each document d has a sequence of N_d words denoted by $d = w_1 \dots . w_{N_d}$. Each word is drawn from a vocabulary V having unique words indexed by $\{1 \dots V\}$. Consider a set of U

users involved in writing the documents in the corpus, where u_d is the author of document d. Consider an ordered set of experience levels $\{e_1, e_2, ...e_E\}$ where each e_i is from a set E, and a set of facets $\{z_1, z_2, ...z_Z\}$ where each z_i is from a set Z of possible facets. Each document d is associated with a rating r and an item i.

At the time t_d of writing the review d, the user u_d has experience level $e_{t_d} \in E$. We assume that her experience level transitions follow a distribution Π with a Markovian assumption and certain constraints. This means the experience level of u_d at time t_d depends on her experience level when writing the previous document at time t_{d-1}.

$\pi_{e_i}(e_j)$ denotes the probability of progressing to experience level e_j from experience level e_i, with the constraint $e_j \in \{e_i, e_i + 1\}$. This means at each instant the user can either stay at her current experience level, or move to the next one.

The experience-level transition probabilities depend on the *rating behavior*, *facet preferences*, and *writing style* of the user. The progression also takes into account the 1) *maturing rate* of u_d modeled by the intensity of her activity in the community, and 2) the *time gaps* between writing consecutive reviews. We incorporate these aspects in a prior for the user's transition rates, γ^{u_d}, defined as:

$$\gamma^{u_d} = \frac{D_{u_d}}{D_{u_d} + D_{avg}} + \lambda(t_d - t_{d-1})$$

D_{u_d} and D_{avg} denote the number of reviews written by u_d and the average number of reviews per user in the community, respectively. Therefore the first term models the user activity with respect to the community average. The second term reflects the time difference between successive reviews. The user experience is unlikely to change from the level when writing the previous review just a few hours or days ago. λ controls the effect of this time difference, and is set to a very small value. Note that if the user writes very infrequently, the second term may go up. But the first term which plays the *dominating* role in this prior will be very small with respect to the community average in an active community, bringing down the influence of the entire prior. Note that the constructed HMM encapsulates all the factors for experience progression outlined in Section 7.1.

At experience level e_{t_d}, user u_d has a Multinomial facet-preference distribution $\theta_{u_d, e_{t_d}}$. From this distribution she draws a facet of interest z_{d_i} for the i^{th} word in her document. For example, a user at a high level of experience may choose to write on the beer "hoppiness" or "story perplexity" in a movie. The word that she writes depends on the facet chosen and the language model for her current experience level. Thus, she draws a word from the multinomial distribution $\phi_{e_{t_d}, z_{d_i}}$ with a symmetric Dirichlet prior δ. For example, if the facet chosen is beer *taste* or movie *plot*, an experienced user may choose to use the words "coffee roasted vanilla" and "visceral", whereas an inexperienced user may use "bitter" and "emotional" resp.

Algorithm 1 describes this generative process for the review; Figure 7.3 depicts it visually in plate notation for graphical models. We use *MCMC* sampling for inference on this model.

7.3.2 Supervision for Rating Prediction

The latent item factors ϕ_i in Equation 7.2 correspond to the latent facets Z in Algorithm 1. Assume that we have some estimation of the latent facet distribution $\phi_{e,z}$ of each document after

Algorithm 1 Supervised Generation Model for a User's Experience, Facets, and Ratings

```
for each facet $z = 1, \ldots, Z$ and experience level $e = 1, \ldots, E$ do
    choose $\phi_{e,z} \sim Dirichlet(\beta)$
for each review $d = 1, \ldots D$ do
    Given user $u_d$ and timestamp $t_d$
    ▷ Current experience level depends on previous level
    Conditioned on $u_d$ and previous experience $e_{t_{d-1}}$, choose $e_{t_d} \sim \pi_{e_{t_{d-1}}}$
    ▷ User's facet preferences at current experience level are influenced by supervision via $\alpha$ - scaled by hyper-parameter $\rho$ controlling influence of supervision.
    Conditioned on supervised facet preference $\alpha_{u_d,e_{t_d}}$ of $u_d$ at experience level $e_{t_d}$ scaled by $\rho$, choose $\theta_{u_d,e_{t_d}} \sim Dirichlet(\rho \times \alpha_{u_d,e_{t_d}})$.
    for each word $i = 1, \ldots, N_d$ do do
        ▷ Facet is drawn from user's experience-based facet interests
        Conditioned on $u_d$ and $e_{t_d}$, choose a facet $z_{d_i} \sim Multinomial(\theta_{u_d}, e_{t_d})$
        ▷ Word is drawn from chosen facet and user's vocabulary at her current experience level.
        Conditioned on $z_{d_i}$ and $e_{t_d}$, choose a word $w_{d_i} \sim Multinomial(\phi_{e_{t_d}}, z_{d_i})$
    ▷ Rating computed via Support Vector Regression with chosen facet proportions as input features to learn $\alpha$
    Choose $r_d \sim F(\langle \alpha_{u_d,e_{t_d}}, \phi_{e_{t_d},z_d} \rangle)$
```

one iteration of MCMC sampling, where e denotes the experience level at which a document is written, and let z denote a latent facet of the document. We also have an estimation of the preference of a user u for facet z at experience level e given by $\theta_{u,e}(z)$.

For each user u, we compute a supervised regression function F_u for the user's numeric ratings with the – currently estimated – experience-based facet distribution $\phi_{e,z}$ of her reviews as input features and the ratings as output.

The learned feature weights $\langle \alpha_{u,e}(z) \rangle$ indicate the user's preference for facet z at experience level e. These feature weights are used to modify $\theta_{u,e}$ to attribute more mass to the facet for which u has a higher preference at level e. This is reflected in the next sampling iteration, when we draw a facet z from the user's facet preference distribution $\theta_{u,e}$ smoothed by $\alpha_{u,e}$, and then draw a word from $\phi_{e,z}$. This sampling process is repeated until convergence.

In any latent facet model, it is difficult to set the hyper-parameters. Therefore, most prior work assume symmetric Dirichlet priors with heuristically chosen concentration parameters. Our approach is to *learn* the concentration parameter α of a *general* (i.e., asymmetric) Dirichlet prior for Multinomial distribution Θ – where we optimize these hyper-parameters to learn user ratings for documents at a given experience level.

7.3.3 Inference

We describe the inference algorithm to estimate the distributions Θ, Φ and Π from observed data. For each user, we compute the conditional distribution over the set of hidden variables E and Z for all the words W in a review. The exact computation of this distribution is intractable. We use *Collapsed Gibbs Sampling* [96] to estimate the conditional distribution for each hidden variable, which is computed over the current assignment for all other hidden variables, and integrating out other parameters of the model.

Let U, E, Z and W be the set of all users, experience levels, facets and words in the corpus.

In the following, i indexes a document and j indexes a word in it.

The joint probability distribution is given by:

$$P(U, E, Z, W, \theta, \phi, \pi; \alpha, \delta, \gamma) = \prod_{u=1}^{U} \prod_{e=1}^{E} \prod_{i=1}^{D_u} \prod_{z=1}^{Z} \prod_{j=1}^{N_{d_u}} \{ \underbrace{P(\pi_e; \gamma^u) \times P(e_i | \pi_e)}_{\text{experience transition distribution}} \times \underbrace{P(\theta_{u,e}; \alpha_{u,e}) \times P(z_{i,j} | \theta_{u,e_i})}_{\text{user experience facet distribution}} \times \underbrace{P(\phi_{e,z}; \delta) \times P(w_{i,j} | \phi_{e_i, z_{i,j}})}_{\text{experience facet language distribution}} \} \tag{7.3}$$

Let $n(u, e, d, z, v)$ denote the count of the word w occurring in document d written by user u at experience level e belonging to facet z. In the following equation, $(.)$ at any position in a distribution indicates summation of the above counts for the respective argument.

Exploiting conjugacy of the Multinomial and Dirichlet distributions, we can integrate out Φ from the above distribution to obtain the posterior distribution $P(Z|U, E; \alpha)$ of the latent variable Z given by:

$$\prod_{u=1}^{U} \prod_{e=1}^{E} \frac{\Gamma(\sum_z \alpha_{u,e,z}) \prod_z \Gamma(n(u, e, ., z, .) + \alpha_{u,e,z})}{\prod_z \Gamma(\alpha_{u,e,z}) \Gamma(\sum_z n(u, e, ., z, .) + \sum_z \alpha_{u,e,z})}$$

where Γ denotes the Gamma function.

Similarly, by integrating out Θ, $P(W|E, Z; \delta)$ is given by

$$\prod_{e=1}^{E} \prod_{z=1}^{Z} \frac{\Gamma(\sum_v \delta_v) \prod_v \Gamma(n(., e, ., z, v) + \delta_v)}{\prod_v \Gamma(\delta_v) \Gamma(\sum_v n(., e, ., z, v) + \sum_v \delta_v)}$$

Let $m_{e_i}^{e_{i-1}}$ denote the number of transitions from experience level e_{i-1} to e_i over *all* users in the community, with the constraint $e_i \in \{e_{i-1}, e_{i-1} + 1\}$. Note that we allow self-transitions for staying at the same experience level. The counts capture the relative difficulty in progressing between different experience levels. For example, it may be easier to progress to level 2 from level 1 than to level 4 from level 3.

The state transition probability depending on the previous state, factoring in the user-specific activity rate, is given by:

$$P(e_i | e_{i-1}, u, e_{-i}) = \frac{m_{e_i}^{e_{i-1}} + I(e_{i-1} = e_i) + \gamma^u}{m_{.}^{e_{i-1}} + I(e_{i-1} = e_i) + E\gamma^u}$$

where $I(.)$ is an indicator function taking the value 1 when the argument is true, and 0 otherwise. The subscript $-i$ denotes the value of a variable excluding the data at the i^{th} position. All the *counts* of transitions exclude transitions to and from e_i, when sampling a value for the current experience level e_i during Gibbs sampling. The conditional distribution for the experience level transition is given by:

$$P(E|U, Z, W) \propto P(E|U) \times P(Z|E, U) \times P(W|Z, E) \tag{7.4}$$

Here the first factor models the rate of experience progression factoring in user activity; the second and third factor models the facet-preferences of user, and language model at a specific level of experience respectively. All three factors combined decide whether the user should stay at the current level of experience, or has matured enough to progress to next level.

In Gibbs sampling, the conditional distribution for each hidden variable is computed based on the current assignment of other hidden variables. The values for the latent variables are sampled repeatedly from this conditional distribution until convergence. In our problem setting we have two sets of latent variables corresponding to E and Z respectively.

We perform Collapsed Gibbs Sampling [96] in which we first sample a value for the experience level e_i of the user for the current document i, keeping all facet assignments Z fixed. In order to do this, we consider two experience levels e_{i-1} and $e_{i-1}+1$. For each of these levels, we go through the current document and all the token positions to compute Equation 7.4 — and choose the level having the highest conditional probability. Thereafter, we sample a new facet for each word $w_{i,j}$ of the document, keeping the currently sampled experience level of the user for the document fixed.

The conditional distributions for Gibbs sampling for the joint update of the latent variables E and Z are given by:

$$\begin{gathered}
\textbf{E-Step 1: } P(e_i = e | e_{i-1}, u_i = u, \{z_{i,j} = z_j\}, \{w_{i,j} = w_j\}, e_{-i}) \propto \\
P(e_i | u, e_{i-1}, e_{-i}) \times \prod_j P(z_j | e_i, u, e_{-i}) \times P(w_j | z_j, e_i, e_{-i}) \propto \\
\frac{m_{e_i}^{e_{i-1}} + I(e_{i-1} = e_i) + \gamma^u}{m_{.}^{e_{i-1}} + I(e_{i-1} = e_i) + E\gamma^u} \times \\
\prod_j \frac{n(u, e, ., z_j, .) + \alpha_{u,e,z_j}}{\sum_{z_j} n(u, e, ., z_j, .) + \sum_{z_j} \alpha_{u,e,z_j}} \times \frac{n(., e, ., z_j, w_j) + \delta}{\sum_{w_j} n(., e, ., z_j, w_j) + V\delta} \\
\textbf{E-Step 2: } \quad P(z_j = z | u_d = u, e_d = e, w_j = w, z_{-j}) \propto \\
\frac{n(u, e, ., z, .) + \alpha_{u,e,z}}{\sum_z n(u, e, ., z, .) + \sum_z \alpha_{u,e,z}} \times \frac{n(., e, ., z, w) + \delta}{\sum_w n(., e, ., z, w) + V\delta}
\end{gathered} \tag{7.5}$$

The proportion of the z^{th} facet in document d with words $\{w_j\}$ written at experience level e is given by:

$$\phi_{e,z}(d) = \frac{\sum_{j=1}^{N_d} \phi_{e,z}(w_j)}{N_d}$$

For each user u, we learn a regression model F_u using these facet proportions in each document as features, along with the user and item biases (refer to Equation 7.2), with the user's item rating r_d as the response variable. Besides the facet distribution of each document, the biases $< \beta_g(e), \beta_u(e), \beta_i(e) >$ also depend on the experience level e.

We formulate the function F_u as Support Vector Regression [64], which forms the M-$Step$ in our problem:

$$\begin{gathered}
\textbf{M-Step: } \min_{\alpha_{u,e}} \frac{1}{2} {\alpha_{u,e}}^T \alpha_{u,e} + C \times \\
\sum_{d=1}^{D_u} (max(0, |r_d - {\alpha_{u,e}}^T < \beta_g(e), \beta_u(e), \beta_i(e), \phi_{e,z}(d) > | - \epsilon))^2
\end{gathered}$$

Table 7.3: Dataset statistics.

Dataset	**#Users**	**#Items**	**#Ratings**
Beer (BeerAdvocate)	33,387	66,051	1,586,259
Beer (RateBeer)	40,213	110,419	2,924,127
Movies (Amazon)	759,899	267,320	7,911,684
Food (Yelp)	45,981	11,537	229,907
Media (NewsTrust)	6,180	62,108	134,407
TOTAL	885,660	517,435	12,786,384

The total number of parameters learned is $[E \times Z + E \times 3] \times U$. Our solution may generate a mix of positive and negative real numbered weights. In order to ensure that the concentration parameters of the Dirichlet distribution are positive reals, we take $exp(\alpha_{u,e})$. The learned α's are typically very small, whereas the value of $n(u, e, ., z, .)$ in Equation 7.5 is very large. Therefore we scale the α's by a hyper-parameter ρ to control the influence of supervision. ρ is tuned using a validation set by varying it from $\{10^0, 10^1 ... 10^5\}$. In the *E-Step* of the next iteration, we choose $\theta_{u,e} \sim Dirichlet(\rho \times \alpha_{u,e})$. We use the LibLinear[2] package for Support Vector Regression.

7.4 Experiments

Setup: We perform experiments with data from five communities in different domains: BeerAdvocate (`beeradvocate.com`) and RateBeer (`ratebeer.com`) for beer reviews, Amazon (`amazon.com`) for movie reviews, Yelp (`yelp.com`) for food and restaurant reviews, and NewsTrust (`newstrust.net`) for reviews of news media. Table 7.3 gives the dataset statistics[3]. We have a total of 12.7 million reviews from 0.9 million users from all of the five communities combined. The first four communities are used for product reviews, from where we extract the following quintuple for our model $< userId, itemId, timestamp, rating, review >$. NewsTrust is a special community, which we discuss in Section 7.5.

For all models, we used the three most recent reviews of each user as withheld test data. All experience-based models consider the *last* experience level reached by each user, and corresponding learned parameters for rating prediction. In all the models, we group *light* users with less than 50 reviews in *training* data into a background model, treated as a single user, to avoid modeling from sparse observations. We do not ignore any user. During the *test* phase for a light user, we take her parameters from the background model. We set $Z = 20$ for BeerAdvocate, RateBeer and Yelp facets; and $Z = 100$ for Amazon movies and NewsTrust which have much richer latent dimensions. For experience levels, we set $E = 5$ for all. However, for NewsTrust and Yelp datasets our model categorizes users to belong to one of *three* experience levels.

[2] http://www.csie.ntu.edu.tw/ cjlin/liblinear

[3] http://snap.stanford.edu/data/, http://www.yelp.com/dataset_challenge/

Table 7.4: MSE comparison of our model versus baselines.

Models	**Beer Advocate**	**Rate Beer**	**News Trust**	**Amazon**	**Yelp**
Our model (most recent experience level)	0.363	0.309	0.373	1.174	1.469
f) Our model (past experience level)	0.375	0.362	0.470	1.200	1.642
e) User at learned rate	0.379	0.336	0.575	1.293	1.732
c) Community at learned rate	0.383	0.334	0.656	1.203	1.534
b) Community at uniform rate	0.391	0.347	0.767	1.203	1.526
d) User at uniform rate	0.394	0.349	0.744	1.206	1.613
a) Latent factor model	0.409	0.377	0.847	1.248	1.560

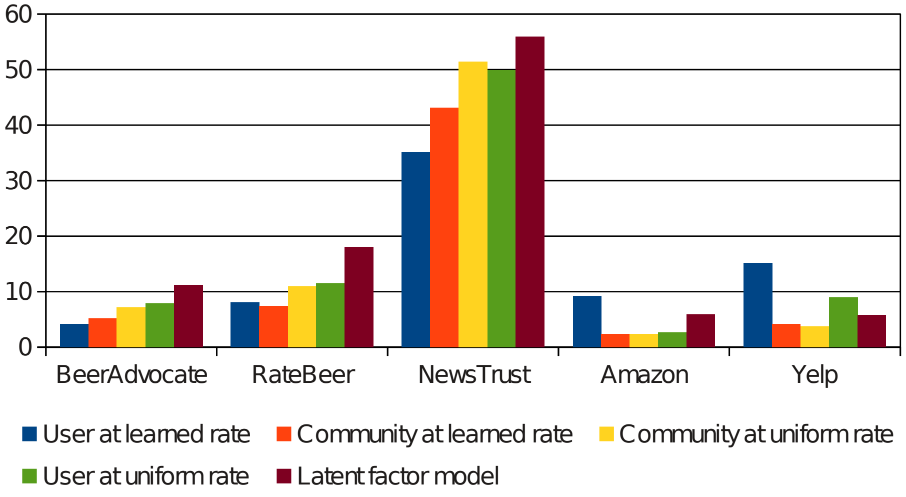


Figure 7.4: MSE improvement (%) of our model over baselines.

7.4.1 Quantitative Comparison

Baselines: We consider the following baselines for our work, and use the available code[4] for experimentation.

a) *LFM*: A standard latent factor recommendation model [147].

b) *Community at uniform rate*: Users and products in a community evolve using a single "global clock" [148][304][302], where the different stages of the community evolution appear at uniform time intervals. So the community prefers different products at different times.

c) *Community at learned rate*: This extends (b) by learning the rate at which the community evolves with time, eliminating the uniform rate assumption.

d) *User at uniform rate*: This extends (b) to consider individual users, by modeling the different stages of a user's progression based on preferences and experience levels evolving over time.

[4] http://cseweb.ucsd.edu/ jmcauley/code/

The model assumes a uniform rate for experience progression.

e) *User at learned rate*: This extends (d) by allowing each user to evolve on a "personal clock", so that the time to reach certain experience levels depends on the user [194].

f) *Our model with past experience level*: In order to determine how well our model captures *evolution of user experience over time*, we consider another baseline where we *randomly sample* the experience level reached by users at some timepoint *previously* in their lifecycle, who may have evolved thereafter. We learn our model parameters from the data up to this time, and again predict the user's most recent three item ratings. Note that this baseline considers textual content of user contributed reviews, unlike other baselines that ignore them. Therefore it is better than vanilla content-based methods, with the notion of past evolution, and is the strongest baseline for our model.

Discussions: Table 7.4 compares the *mean squared error (MSE)* for rating predictions, generated by our model versus the six baselines. Our model consistently outperforms all baselines, reducing the MSE by ca. 5 to 35%. Improvements of our model over baselines are statistically significant at p-value < 0.0001.

Our performance improvement is most prominent for the NewsTrust community, which exhibits strong language features, and topic polarities in reviews. The lowest improvement (over the best performing baseline in any dataset) is achieved for Amazon movie reviews. A possible reason is that the community is very diverse with a very wide range of movies and that review texts heavily mix statements about movie plots with the actual review aspects like praising or criticizing certain facets of a movie. The situation is similar for the food and restaurants case. Nevertheless, our model always wins over the best baseline from *other* works, which is typically the "user at learned rate" model.

Evolution effects: We observe in Table 7.4 that our model's predictions degrade when applied to the users' *past* experience level, compared to their *most recent* level. This signals that the model captures user evolution past the previous timepoint. Therefore the last (i.e., most recent) experience level attained by a user is most informative for generating new recommendations.

7.4.2 Qualitative Analysis

Salient words for facets and experience levels: We point out typical word clusters, with *illustrative* labels, to show the variation of language for users of different experience levels and different facets. Tables 7.2 and 7.5 show salient words to describe the beer facet *taste* and movie facets *plot* and *narrative style*, respectively – at different experience levels. Note that the facets being latent, their labels are merely our interpretation. Other similar examples can be found in Tables 7.1 and 7.7.

BeerAdvocate and RateBeer are very focused communities; so it is easier for our model to characterize the user experience evolution by vocabulary and writing style in user reviews. We observe in Table 7.5 that users write more descriptive and *fruity* words to depict the beer taste as they become more experienced.

For movies, the wording in reviews is much more diverse and harder to track. Especially for blockbuster movies, which tend to dominate this data, the reviews mix all kinds of aspects. A better approach here could be to focus on specific kinds of movies (e.g., by genre or production

Table 7.5: Experience-based facet words for the *illustrative* beer facet *taste*.

Experience Level 1: drank, bad, maybe, terrible, dull, shit
Experience Level 2: bottle, sweet, nice hops, bitter, strong light, head, smooth, good, brew, better, good
Expertise Level 3: sweet alcohol, palate down, thin glass, malts, poured thick, pleasant hint, bitterness, copper hard
Experience Level 4: smells sweet, thin bitter, fresh hint, honey end, sticky yellow, slight bit good, faint bitter beer, red brown, good malty, deep smooth bubbly, damn weak
Experience Level 5: golden head lacing, floral dark fruits, citrus sweet, light spice, hops, caramel finish, acquired taste, hazy body, lacing chocolate, coffee roasted vanilla, creamy bitterness, copper malts, spicy honey

Table 7.6: Distribution of users at different experience levels.

Datasets	**e=1**	**e=2**	**e=3**	**e=4**	**e=5**
BeerAdvocate	0.05	0.59	0.19	0.10	0.07
RateBeer	0.03	0.42	0.35	0.18	0.02
NewsTrust	-	-	0.15	0.60	0.25
Amazon	-	0.72	0.13	0.10	0.05
Yelp	-	-	0.30	0.68	0.02

studios) that may better distinguish experienced users from amateurs or novices in terms of their refined taste and writing style.

MSE for different experience levels: We observe a weak trend that the MSE decreases with increasing experience level. Users at the highest level of experience almost always exhibit the lowest MSE. So we tend to better predict the rating behavior for the most mature users than for the remaining user population. This in turn enables generating better recommendations for the "connoisseurs" in the community.

Experience progression: Figure 7.5 shows the proportion of reviews written by community members at different experience levels right before advancing to the next level. Here we plot users with a minimum of 50 reviews, so they are certainly not "amateurs". A large part of the community progresses from level 1 to level 2. However, from here only few users move to higher levels, leading to a skewed distribution. We observe that the majority of the population stays at level 2.

User experience distribution: Table 7.6 shows the number of users per experience level in each domain, for users with > 50 reviews. The distribution also follows our intuition of a highly skewed distribution. Note that almost all users with < 50 reviews belong to levels 1 or 2.

Language model and facet preference divergence: Figure 7.6b and 7.6c show the KL divergence for facet-preference and language models of users at different experience levels, as

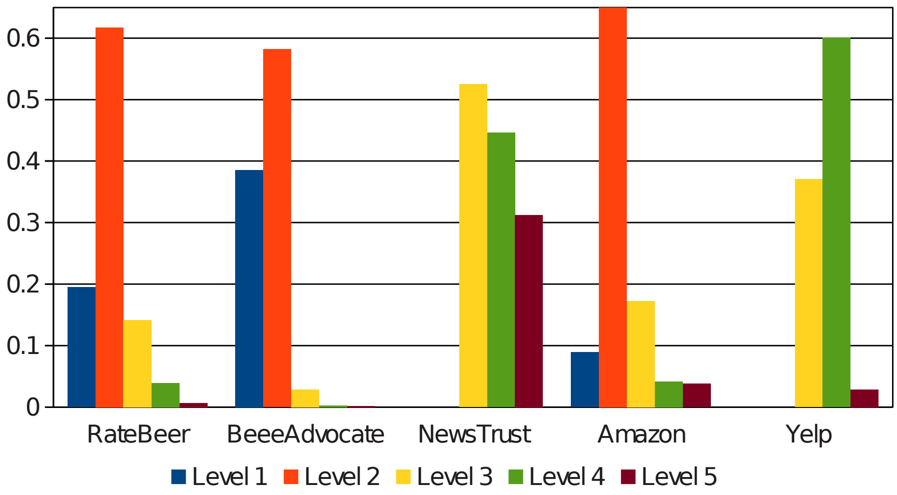


Figure 7.5: Proportion of reviews at each experience level of users.

computed by our model. The facet-preference divergence increases with the gap between experience levels, but not as *smooth* and prominent as for the language models. On one hand, this is due to the complexity of *latent* facets vs. *explicit* words. On the other hand, this also affirms our notion of grounding the model on *language*.

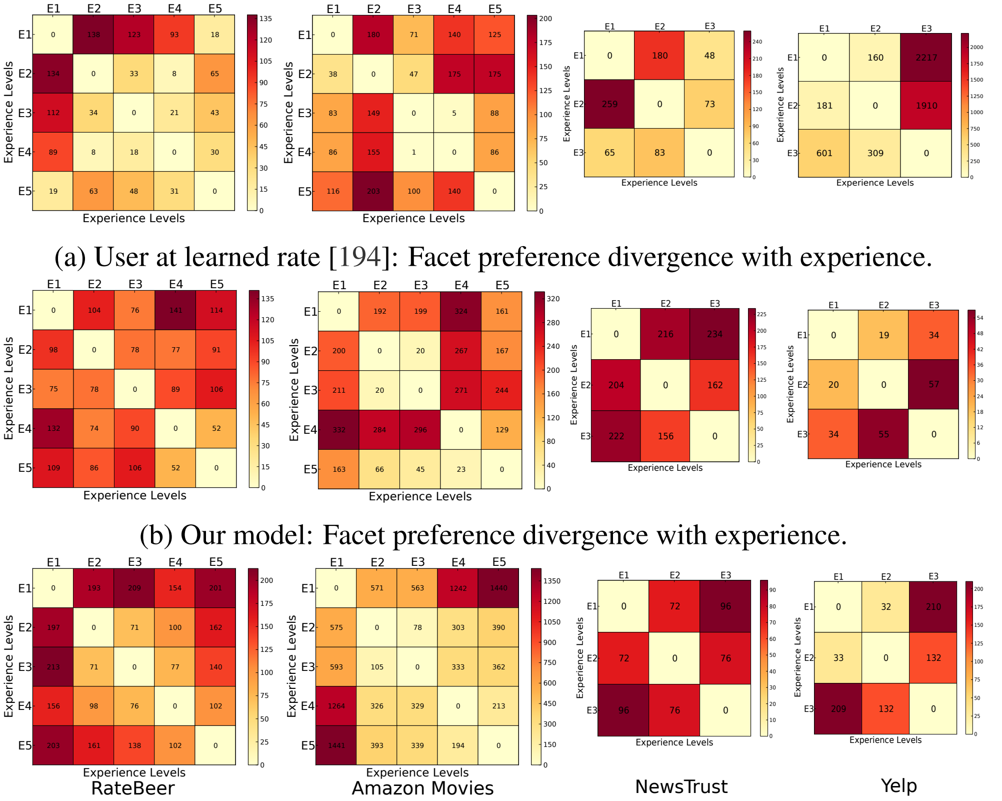
Baseline model divergence: Figure 7.6a shows the facet-preference divergence of users at different experience levels computed by the baseline model "user at learned rate" [194]. The contrast between the heatmaps of our model and the baseline is revealing. The increase in divergence with increasing gap between experience levels is very *rough* in the baseline model, although the trend is obvious.

7.5 Use-Case Study

So far we have focused on traditional item recommendation for items like beers or movies. Now we switch to a different kind of items - newspapers and news articles - tapping into the NewsTrust online community (`newstrust.net`). NewsTrust features news stories posted and reviewed by members, many of whom are professional journalists and content experts. Stories are reviewed based on their objectivity, rationality, and general quality of language to present an unbiased and balanced narrative of an event. The focus is on *quality journalism*.

In our framework, each story is an item, which is rated and reviewed by a user. The facets are the underlying topic distribution of reviews, with topics being *Healthcare, Obama Administration, NSA*, etc. The facet preferences can be mapped to the (political) polarity of users in the news community.

Recommending News Articles: Our first objective is to recommend news to readers catering to their facet preferences, viewpoints, and experience. We apply our joint model to this task, and compare the predicted ratings with the ones observed for withheld reviews in the NewsTrust community. The mean squared error (MSE) results are reported in Table 7.4 in Section 7.4.



	E1	E2	E3	E4	E5
E1	0	138	123	93	18
E2	134	0	33	8	65
E3	112	34	0	21	43
E4	89	8	18	0	30
E5	19	63	48	31	0

	E1	E2	E3	E4	E5
E1	0	180	71	140	125
E2	38	0	47	175	175
E3	83	149	0	5	88
E4	86	155	1	0	86
E5	116	203	100	140	0

	E1	E2	E3
E1	0	180	48
E2	259	0	73
E3	65	83	0

	E1	E2	E3
E1	0	160	2217
E2	181	0	1910
E3	601	309	0

(a) User at learned rate [194]: Facet preference divergence with experience.

	E1	E2	E3	E4	E5
E1	0	104	76	141	114
E2	98	0	78	77	91
E3	75	78	0	89	106
E4	132	74	90	0	52
E5	109	86	106	52	0

	E1	E2	E3	E4	E5
E1	0	192	199	324	161
E2	200	0	20	267	167
E3	211	20	0	271	244
E4	332	284	296	0	129
E5	163	66	45	23	0

	E1	E2	E3
E1	0	216	234
E2	204	0	162
E3	222	156	0

	E1	E2	E3
E1	0	19	34
E2	20	0	57
E3	34	55	0

(b) Our model: Facet preference divergence with experience.

	E1	E2	E3	E4	E5
E1	0	193	209	154	201
E2	197	0	71	100	162
E3	213	71	0	77	140
E4	156	98	76	0	102
E5	203	161	138	102	0

RateBeer

	E1	E2	E3	E4	E5
E1	0	571	563	1242	1440
E2	575	0	78	303	390
E3	593	105	0	333	362
E4	1264	326	329	0	213
E5	1441	393	339	194	0

Amazon Movies

	E1	E2	E3
E1	0	72	96
E2	72	0	76
E3	96	76	0

NewsTrust

	E1	E2	E3
E1	0	32	210
E2	33	0	132
E3	209	132	0

Yelp

(c) Our model: Language model divergence with experience.

Figure 7.6: Facet preference and language model KL divergence with experience.

Table 7.7: Salient words for the *illustrative* NewsTrust topic *US Election* at different experience levels.

Level 1: bad god religion iraq responsibility
Level 2: national reform live krugman questions clear jon led meaningful lives california powerful safety impacts
Level 3: health actions cuts medicare nov news points oil climate major jobs house high vote congressional spending unemployment strong taxes citizens events failure

Table 7.8: Performance on identifying experienced users.

Models	F_1	$NDCG$
User at learned rate [194]	0.68	0.90
Our model	0.75	0.97

Table 7.7 shows salient examples of the vocabulary by users at different experience levels on the topic *US Election*.

Identifying Experienced Users: Our second task is to find experienced members of this community, who have potential for being *citizen journalists*. In order to find how good our model predicts the experience level of users, we consider the following as ground-truth for user experience. In NewsTrust, users have *Member Levels* calculated by the NewsTrust staff based on community engagement, time in the community, other users' feedback on reviews, profile transparency, and manual validation. We use these member levels to categorize users as *experienced* or *inexperienced*. This is treated as the ground truth for assessing the prediction and ranking quality of our model and the baseline "user at learned rate" model [194]. Table 7.8 shows the F_1 scores of these two competitors. We also computed the *Normalized Discounted Cumulative Gain (NDCG)* [128] for the ranked lists of users generated by the two models. NDCG gives geometrically decreasing weights to predictions at various positions of ranked list: $NDCG_p = \frac{DCG_p}{IDCG_p}$ where $DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$. Here, rel_i is the relevance (0 or 1) of a result at position i.

As Table 7.8 shows, our model clearly outperforms the baseline model on both F_1 and $NDCG$.

7.6 Related Work

State-of-the-art recommenders based on collaborative filtering [147][149] exploit user-user and item-item similarities by latent factors. Explicit user-user interactions have been exploited in trust-aware recommendation systems [101][296]. The temporal aspects leading to bursts in item popularity, bias in ratings, or the evolution of the entire community as a whole is studied in [148][304][302]. Other papers have studied temporal issues for anomaly detection [103], detecting changes in the social neighborhood [185] and linguistic norms [54]. However, none of these prior work has considered the evolving experience and behavior of individual users.

The recent work[194], which is one of our baselines, modeled the influence of rating behavior on evolving user experience. However, it ignores the vocabulary and writing style of users in reviews, and their natural *smooth* temporal progression. In contrast, our work considers the review texts for additional insight into facet preferences and *smooth* experience progression.

Prior work that tapped user review texts focused on other issues. Sentiment analysis over reviews aimed to learn latent topics [175], latent aspects and their ratings [155][293], and user-user interactions [296]. [193][290] unified various approaches to generate user-specific ratings of reviews. [208] further leveraged the author writing style. However, all of these prior approaches operate in a static, snapshot-oriented manner, without considering time at all.

From the modeling perspective, some approaches learn a document-specific discrete rating [175][236], whereas others learn the facet weights outside the topic model (e.g., [155, 193, 208]). In order to incorporate continuous ratings, [27] proposed a complex and computationally expensive Variational Inference algorithm, and [199] developed a simpler approach using Multinomial-Dirichlet Regression. The latter inspired our technique for incorporating supervision.

7.7 Conclusion

Current recommender systems do not consider user experience when generating recommendations. In this paper, we have proposed an experience-aware recommendation model that can adapt to the changing preferences and maturity of users in a community. We model the *personal evolution* of a user in rating items that she will appreciate at her current maturity level. We exploit the coupling between the *facet preferences* of a user, her *experience*, *writing style* in reviews, and *rating behavior* to capture the user's temporal evolution. Our model is the first work that considers the progression of user experience as expressed in the text of item reviews.

Our experiments – with data from domains like beer, movies, food, and news – demonstrate that our model substantially reduces the mean squared error for predicted ratings, compared to the state-of-the-art baselines. This shows our method can generate better recommendations than those models. We further demonstrate the utility of our method in a use-case study about identifying experienced members in the NewsTrust community who can be potential citizen journalists.

Part IV

Identifying Fraud on Social Media

Neil Shah, Hemank Lamba, Alex Beutel, Christos Faloutsos. ”The many faces of link fraud”. 2017 IEEE International Conference on Data Mining (ICDM), 2017.

CHAPTER 8

UNDERSTANDING LINK FRAUD SERVICES

Most past work on social network link fraud detection tries to separate genuine users from fraudsters, implicitly assuming that there is only one type of fraudulent behavior. But is this assumption true? And, in either case, what are the characteristics of such fraudulent behaviors? In this work, we set up *honeypots*, (“dummy” social network accounts), and buy fake followers (after careful IRB approval). We report the signs of such behaviors including oddities in local network connectivity, account attributes, and similarities and differences across fraud providers. Most valuably, we discover and characterize *several* types of fraud behaviors. We discuss how to leverage our insights in practice by engineering strongly performing entropy-based features and demonstrating high classification accuracy. Our contributions are (a) *instrumentation*: we detail our experimental setup and carefully engineered data collection process to scrape Twitter data while respecting API rate-limits, (b) *observations on fraud multimodality*: we analyze our honeypot fraudster ecosystem and give surprising insights into the multifaceted behaviors of these fraudster types, and (c) *features*: we propose novel features that give strong (>*0.95* precision/recall) discriminative power on ground-truth Twitter data.

What are the characteristics of fraudulent accounts in online social networks? Understanding the behavior and actions of fraudsters is paramount to building effective anti-fraud algorithms. While previous works in social network fraud detection have primarily focused on leveraging signature properties of fraudsters including temporally synchronized behavior [22], excessively dense [233] and oddly distributed [253] graph connectivity, uncommon account names [75] and spammy links [95], our work focuses on establishing the veracity and applicability of these assumptions. In doing so, we ask: do all fraudsters share the same signature behavior, or are there multiple signatures? Since fraud detection is an adversarial setting in which fraudsters are constantly adapting to in-place detection mechanisms, it is important to constantly monitor and evaluate the strategies that fraudsters are employing to profitably perform ingenuine actions to better inform future detection mechanisms.

We focus on one particular setting of social network fraud called *link fraud* which involves the use of fake, *sockpuppet* accounts to create links, or graph connections, which represent followership or support of target, customer entities. Fake links artificially inflate the follower count of customer accounts, making them appear more popular than they actually are. These fake links are deceptive to authentic users and hinder the performance of machine learning algorithms which rely on authentic user input to recommend relevant and useful content to their userbase.

To study the behavior of these fake follower accounts, we employ the use of *honeypots*, or dummy accounts on which we solicit fake Twitter followers sourced from various fraud service providers. Honeypots enable us to have an clear signal of fake follower activity which is not tainted by follows from real accounts. Upon setting up the honeypot accounts and purchasing fake followers, we instrument a number of carefully engineered tracking scripts which poll Twitter API to store details including account relationships and attributes over a period of time. This allows us to collect a rich representation of the fraudster ecosystem which we subsequently analyze.

In this work, we make and explore the following key observation:

Key Insight 1 (Fraud Multimodality)**.** *There are multiple types of link fraud which exhibit notably different network structures and patterns in account attribute settings.*

Specifically, we focus on studying and characterizing the network connectivity properties and attribute distributions which are exhibited by fake followers involved in these different types of fraud. We detail a number of further observations on how these types of behavior induce different, odd network structures and suspicious patterns in account attributes. Figure 8.1 shows the contrast in follower connectivity of a genuine account versus two distinct types of fraudsters. Through our analysis, we additionally engineer strong features which enable us to discriminate these fraudulent users from genuine ones using novel (first-order) follower entropy features.

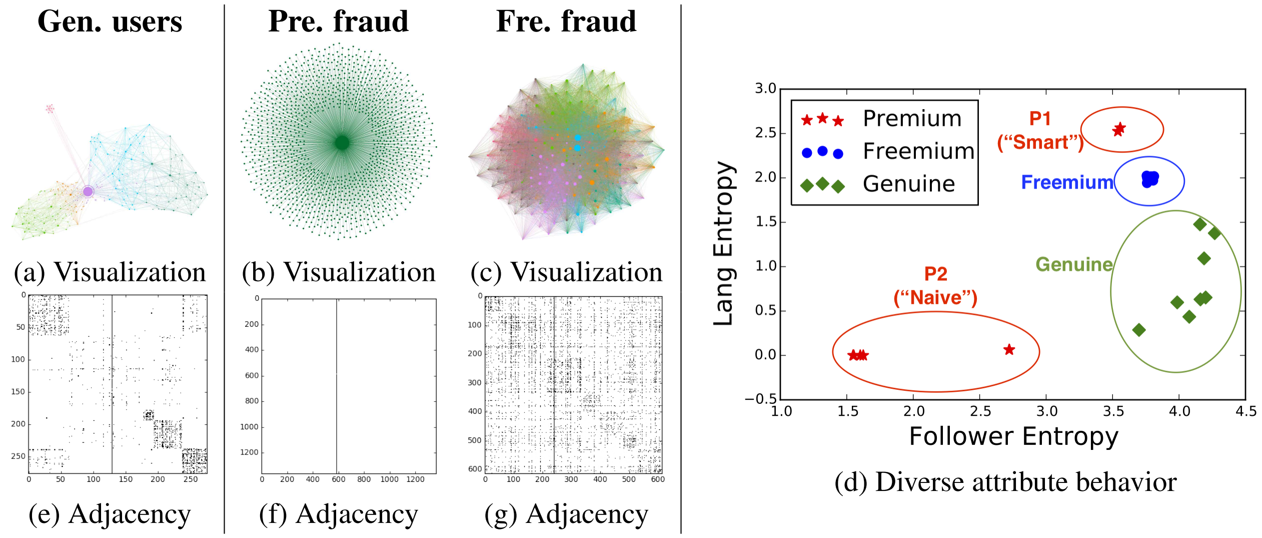
Summarily, our work offers the following notable contributions:

- **Instrumentation**: We detail our experimental setup and data scraping tools which gather a wealth of Twitter user information while respecting API rate limits.
- **Observations on Fraud Multimodality**: We discover that link fraud is not unimodal and instead has multiple types, and identify and characterize two such types: *freemium* and *premium*, with the possibility of more.
- **Features**: Based on the above observations, we carefully engineer novel, entropy-based features which allow us to accurately discern fraudsters from genuine users in our ground-truth Twitter dataset with near-perfect F1-score.

8.1 Related Work

We categorize related work into two categories: underground market studies and fraud detection approaches.

Underground Markets: Prior works have shown the use of fake accounts for followers in social media [277], phone-verified email accounts [278], Facebook likes [22], etc. These accounts are often used to spread spam [79, 95] and misinformation [105, 106]. [227] estimates that the fake follower market produces $360 million per year. Recently, several works have studied the existence of underground online markets where these fraudulent actions can be purchased – [207, 292] explore underground markets providing fake content, reviews and solutions to security



(a) Visualization (b) Visualization (c) Visualization

(e) Adjacency (f) Adjacency (g) Adjacency

(d) Diverse attribute behavior

Figure 8.1: **Freemium (Fre) and premium (Pre) fraud types have different local network structure and account attributes compared to genuine behavior.** Nodes are colored by modularity class, and sized proportional to in-degree in (a)-(c). The associated, reordered adjacency matrices are shown in (e)-(g) – the vertical line in each spyplot indicates the the central node. Notice the block community structure in genuine followers compared to the star structure for premium and near-clique structure for freemium followers. (d) shows differences in attribute (language and follower) entropy over the various behaviors, showing how fraud patterns skew attribute distributions away from genuine ones.

mechanisms. [277] studies several fraud providers over time and describes trends in pricing, account names and IP diversity. [268] compares growth rates of accounts with legitimate and fraudulent followers. [9] observes the varying retention and reliability of various fraud providers. Comparatively, our work is the first to identify major social graph differences between fraud types and across providers, and propose novel entropy-based features for capturing these behaviors.

Fraud Detection: [20, 170] use profile features to detect spammers on Twitter. [267] passively analyzes accounts with promiscuous following behavior and builds a classifier using profile and messaging features. [38, 309] aim to find fake accounts in social networks via a generative stochastic model and a random-walk based method respectively – both assume small cuts between fake and genuine nodes. [22, 40] use graph-traversal based methods to find users with temporally synchronized actions on Facebook. [130, 233, 253] propose spectral methods which identify dense or odd graph structures indicative of fraud.

8.2 Know Thy Enemy: Characterizing Link Fraud

In this section, we discuss some preliminaries about instrumentation, data collection and relevant metrics, and next illustrate numerous insights about network connectivity and account attributes of link fraudsters.

Table 8.1: Honeypot account summaries.

Service	**Type**	**Cost**	**Followers bought**	**Followers delivered**	**Followers remaining**
`fastfollowerz`	Premium	\$19	1000	1060 1060	1059 1059
`intertwitter`	Premium	\$14	1000	1099 1102	977 974
`devumi`	Premium	\$19	1000	1360 1354	1358 1354
`twitterboost`	Premium	\$12	1000	1361 1350	1361 1350
`plusfollower`	Freemium	£9.99	1000	1094 1078	748 737
`hitfollow`	Freemium	£9.99	1000	926 937	623 638
`newfollow`	Freemium	£9.99	1000	884 883	600 589
`bigfolo`	Freemium	£9.99	1000	872 865	594 577

8.2.1 Setup and Data Collection

We first discuss how we identified and purchased followers from target fraud service providers, and next detail the scraping task, followed by preliminaries.

Purchasing Fake Followers

There are a number of different fraud service providers easily accessible and available on the web. We begin by identifying these services so we can purchase fake followers from them. To identify these services, we used Google search and queried using keywords such as "buy Twitter followers." Combining the search results, we obtained a list of websites which claim to provide these services.

From surveying the websites on this list, we notice there are several prevalent models of service – we categorize these into two frameworks: *premium* and *freemium*. Premium services offer customers multiple tiers of follower counts (1K, 5K, 10K, etc.) for various amounts of money and ask only for the customer's Twitter username and a form of payment. Freemium services offer both a paid option as in premium services, but additionally offer a free option which does not ask the user for money, but instead requires the user to provide their Twitter login details to the service. In return for these details, the services promise to direct a small number of followers to the account.

We next setup a pool of honeypot accounts by repeating the Twitter account creation process a number of times using monikers from online screenname generators. We found that to create a sizeable pool of honeypots, we needed to distribute the account creation over several IPs in order to avoid phone verification prompts. Upon setting up the pool of honeypots, we purchased basic follower packages from several premium and paid freemium services, avoiding rarely used ones with low Alexa rank. Summarily, we bought 1K followers from 8 different services (4 freemium, 4 premium) to 2 honeypot accounts per service. We chose to purchase 2 honeypot accounts per service instead of only 1 in order to examine the overlap dynamics of fake links to multiple customers. The final list of the services we used, service types, costs and their follower counts are summarized Table 8.1. Honeypots were created on the same day, and follower purchases were all done at the same time. Furthermore, the honeypots attracted no followers by themselves prior to the purchases. As a result, we posit that all followers of the honeypots are fake.

Instrumentation Details

Reproducibility: Code available at `https://goo.gl/qMBWim`.

We use the REST API to scrape data relevant to our operation from Twitter. As the API heavily rate-limits various data resource types, it is only feasible to extract a limited amount of information as an end-user. Prior to purchasing fake followers, we start a number of Python scripts which poll the API and insert data into a Postgres database:

Honeypot account details: Every hour, we collect public details for each honeypot Twitter account including number of friends and followees, number of favorites, number of Tweets, language, etc.

Honeypot account follower IDs: Every 12 hours, we collect the list of follower IDs for each honeypot. Since the honeypots were created with empty profiles, we can safely assume that all followers to these accounts were fraudulent and purchased.

Honeypot account follower details: Every day, we extract public details for each of the accounts in the honeypot follower list.

Honeypot account followers' friends/followers IDs: Every day, we collect the list of friend and follower IDs of the honeypot followers to examine their other connectivity.

Honeypot account followers' friends/followers details: Every 3 days, we extract public details for each of the friends and followers of the honeypot followers to gain more information about them.

Account details requests are limited to 15 requests per 15 minute window, and each request returns details for up to 100 accounts. Similarly, ID list requests are limited to 180 requests per 15 minute window, and each request returns up to 5000 account IDs. Hence, it is relatively easy to scrape the first-order honeypot account follower IDs and details without exceeding the rate limit, but collecting details for the second-order followers is a bottleneck. Since the number of nodes to collect information for can explode substantially even at the second-order, we limit collection to ¡100K friends and followers for each of the given follower of the honeypot account. We determine periodicity values empirically using back-of-the-envelope calculations. While this data could be collected slowly using a single Twitter API key, we speed up the process by using multiple keys and cycling keys upon resource exhaustion.

Preliminaries

In the remainder of our work, we conduct analysis on two types of networks: the *ego network* and *boomerang network*.

Ego network: An ego network (or egonet) traditionally consists of a central node called the ego, as well as the neighboring nodes and the relationships (edges) between them. Egonets can essentially be considered as a local graphical representation of a node within the context of the broader, global graph and depict how the surrounding nodes are connected. For our purposes, we examine *per-service* egonets, where we consider the union of the individual egonets of both honeypot accounts per service. Thus, in our case, each per-service egonet is actually comprised by 2 egos (the honeypot accounts), the union of both honeypots' neighboring nodes (the purchased, fake followers) and the relationships between them. The per-service egonet representation allows us to both individually study the *per-honeypot* egonets as well as any interactions between them. That is, if the two honeypots for each service have distinct sets of neighboring nodes, then their per-honeypot egonets will also be distinct. Conversely, if any nodes are neighbors of both honeypots, the associated per-honeypot egonets will be conjoined. Various levels of overlap suggest differences with regards to how services reuse accounts to deliver fake links.

Boomerang network: Drawing conclusions from per-service egonet analysis can be deceiving in the sense that while it does give insights into the *internal* relationships between the fake followers and honeypots, it does not consider the *external* relationships formed by the fake followers. As such, it is unable to give us a full perspective on the utilization of these fake followers. In order to gain the requisite perspective, we conduct analysis of the proposed boomerang network. We define the per-service boomerang network to be comprised of the per-service egonet *in addition to* the out-links of the follower nodes – the structure is reminiscent of a boomerang, in that it is comprised of the nodes "1 step back and 1 step forward" with respect to the honeypot account. Thus, the per-service boomerang network gives us an additional layer of information on top of the per-service egonet: connections to the other accounts followed by the honeypot's fake followers.

We further use the *density*, *bipartite density*, *transitivity* and *reciprocity* metrics to summarize and describe network structure, and *overlap coefficient* and *multiple systems estimation* (MSE) to characterize network overlap.

Density: We define density as

$$\frac{\#\text{edges}}{\#\text{nodes} \cdot (\#\text{nodes} - 1)}$$

Density represents the fraction of existing to possible total edges, with density 1 indicating a complete graph.

Bipartite density: We define bip. density between sets $\mathcal{A}$ and $\mathcal{B}$ as

$$\frac{\#\text{edges between } \mathcal{A} \text{ and } \mathcal{B}}{(\#\text{nodes in } \mathcal{A}) \cdot (\#\text{nodes in } \mathcal{B})}$$

Bipartite density captures the fraction of existing to possible edges between two sets of nodes, with bipartite density 1 indicating a complete bipartite graph.

Transitivity: We define transitivity as

$$\frac{3 \cdot \#\text{triangles}}{\#\text{connected triples}}$$

Transitivity denotes the degree of triadic closure, with transitivity 1 indicating that all connected triples of nodes are also triangles.

Reciprocity: We define reciprocity as

$$\frac{\#\text{bidirectional edges}}{\#\text{edges}}$$

Reciprocity conveys the relative frequency of bidirectional edges, with reciprocity 1 indicating that all edges are bidirectional.

Overlap coefficient: We define overlap coef. between $\mathcal{A}$ and $\mathcal{B}$ as

$$\frac{|\mathcal{A} \,\cap\, \mathcal{B}|}{min(|\mathcal{A}|, |\mathcal{B}|)}$$

Overlap coefficient indicates the proportion of members that overlap between sets, with overlap coefficient 1 indicating that $\mathcal{A} \subseteq \mathcal{B}$ or $\mathcal{B} \subseteq \mathcal{A}$ and 0 indicating $\mathcal{A} \cap \mathcal{B} = \emptyset$.

Multiple systems estimation: We use MSE to estimate population size from two randomly sampled sets $\mathcal{A}$ and $\mathcal{B}$ as

$$\frac{|\mathcal{A}| \,\cdot\, |\mathcal{B}|}{|\mathcal{A} \,\cap \mathcal{B}|}$$

Intuitively, if $\mathcal{A}$ and $\mathcal{B}$ have low overlap, the total population size is much larger than if they have high overlap.

Upon shifting our discussion to account attributes distributions, we use *entropy* as a means to capture distributional skew.

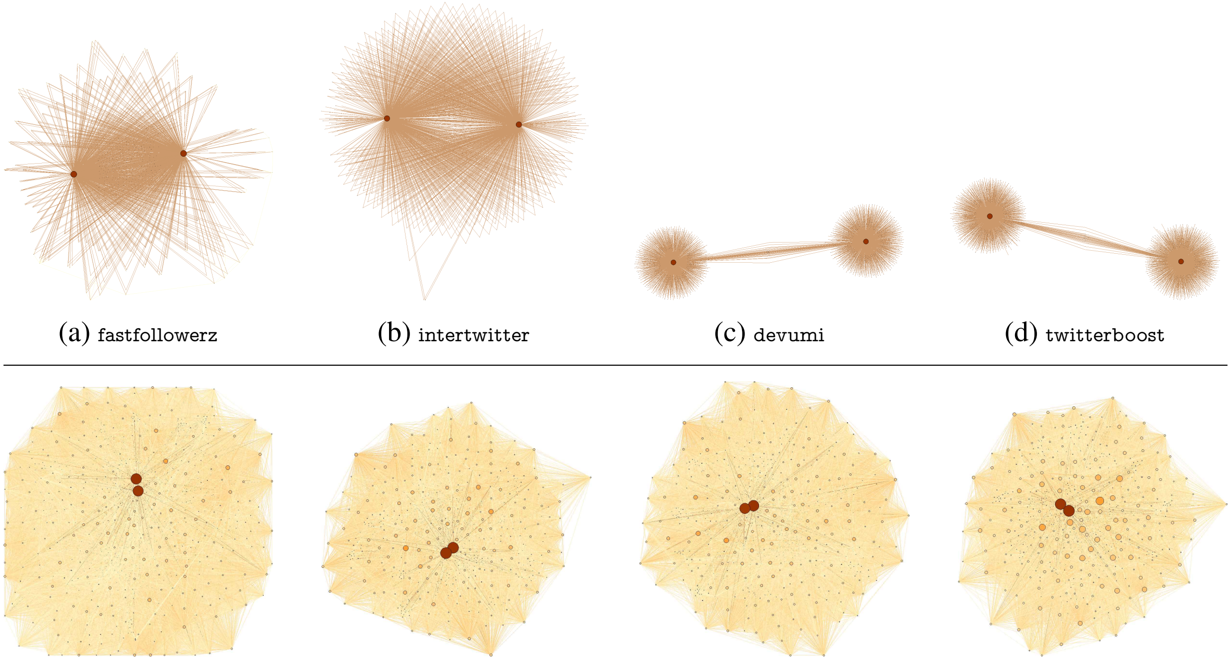
Entropy: We define entropy for a distribution X with n outcomes $(x_1 \ldots x_n)$ as

$$-\sum_{i=1}^{n} P(x_i) \cdot \log_2 P(x_i)$$

Entropy measures the unpredictability of a distribution in bits of information, with entropy of 0 bits indicating concentration of 100% probability on a single outcome, and entropy of $\log_2 n$ bits indicating uniform distribution of probability between n outcomes.

8.2.2 Network Observations

We first focus on studying the local network properties of fraudulent accounts. Targeting oddities in network connectivity is a central theme in many link fraud detection approaches, as the mission constraints of delivering fake links to customers necessarily affects graph structure. But what are these changes? In this section, we leverage social network analysis tools to characterize effects of fraud on the surrounding network structure, and show the similarities and differences between premium and freemium fraud. We detail analyses on two types of induced subgraphs: the ego network and more expansive boomerang network.



(a) fastfollowerz (b) intertwitter (c) devumi (d) twitterboost

(e) plusfollower (f) newfollow (g) hitfollow (h) bigfolo

Figure 8.2: **Premium fraudsters (top) form overlapping stars whereas freemium ones (bottom) form dense, near-cliques.** Subplots show per-service egonets with honeypots in dark-red – darker color and larger size indicates higher in-degree.

Ego Network Patterns

Figure 8.2 shows the per-service egonets for each of the 8 providers, with increased node size and darkness corresponding to higher in-degree. The honeypots (egos) are the two large and dark orange colored nodes in each subfigure. Cursory analysis reveals a notable difference in egonet network structure between freemium and premium providers. We see that the premium egonets (first row) have a star/bipartite structure: each honeypot node is the hub of a star, and the satellite nodes overlap and are disconnected. Conversely, freemium egonets have denser, near-clique type structure which suggests denser connectivity between the neighboring nodes.

The statistics for premium service egonets in Table 8.2 (top) further lend credence to the visual differences we observe from Figure 8.2, giving us the following insight:

Insight 1 (Egonet Sparsity)**.** *Premium fake followers rarely follow each other, resulting in sparse egonet structure. Freemium fake followers have dense egonet structure.*

This is substantiated by the low density and node to edge ratios across premium providers. Of these, fastfollowerz and intertwitter have an order of magnitude greater density than devumi and twitterboost. This is substantiated by the 1:2 node to edge ratio in the former 2 providers as compared to the near 1:1 ratios of the latter 2. fastfollowerz and intertwitter also have marginally higher transitivity values compared to the 0 transitivity of devumi and twitterboost, indicating that the former 2 have few triangles between the fake follower nodes whereas the latter 2 have none. We also observe no reciprocal links in these providers, indicating only one-way relationships.

Table 8.2: Egonet summary statistics.

	Service	# Nodes	# Edges	Density	Transitivity	Reciprocity
Premium	fastfollowerz	1,066	2,289	.002	.001	.000
	intertwitter	1,051	2,003	.002	.00006	.000
	devumi	2,681	2,712	.0003	.000	.000
	twitterboost	2,680	2,711	.0004	.000	.000
Freemium	plusfollower	920	51,868	.061	.288	.411
	newfollow	755	37,052	.065	.294	.408
	hitfollow	782	41,879	.068	.305	.416
	bigfolo	749	36,043	.064	.294	.413

Conversely, the freemium statistics in Table 8.2 (bottom) support that freemium fake followers have dense egonet structures. Freemium providers are an order of magnitude denser than the densest premium egonets – all 4 providers have 6-7% density. While not shown in interest of space, the per-honeypot egonets were each found to have an even higher 11-14% density individually. The 1:50 node to edge ratios substantiate this high density. We also notice that transitivity values are much higher for freemium providers, suggesting that an unusually high 28-30% of wedges are also triangles. Given that density and transitivity are equal in random graphs, the freemium egonets do not appear to be random, but are likely composed of dense subregions which are themselves sparsely connected. The link structure reflects how freemium providers trade follows between accounts (random partitions, biased selection, account similarity, etc.) Furthermore, all 4 providers have similar, high reciprocity of 40-42% suggesting frequent "follow-back" behavior.

Rationale: The freemium services accumulates a pool of free accounts, and hence trading follows enables each free user to gain some followers. As a result, such behavior creates a denser subgraph, but are also used by providers to deliver the follower demands of paid customers and turn a profit. Comparatively, premium providers are unable to use free users' accounts and must create fake accounts.

These insights pose an interesting question: as we expect fraudsters to act in a manner that maximizes profit, *what motivates the differences in structure between freemium and premium providers?* We propose an answer: If we consider that each account has a budget of edges it can create without being suspended, it seems that premium providers greatly underutilize accounts compared to freemium ones. This is because for fraudsters, delivering more links while avoiding suspension is strictly better as it means that they can either serve more customers or artificially inflate their own popularity.

Boomerang Network Patterns

Figure 8.3 shows 2 boomerang networks, one for bigfolo and twitterboost, each representative of a different fraud strategy. Again, honeypot accounts are amongst the large, dark nodes with high in-degree, and the lighter, smaller nodes are fake followers or their friends. Note that the layout clusters nodes based on similar linkage, so groups of nodes visually close share connectiv-

Table 8.3: Boomerang network summary statistics.

	Service	# Nodes	# Edges	Bip. Density
Premium	fastfollowerz	40,486	491,458	.012
	intertwitter	176,921	2,383,251	.013
	devumi	67,893	2,495,586	.014
	twitterboost	68,297	2,474,759	.014
Freemium	plusfollower	646,901	1,352,253	.002
	newfollow	616,824	1,221,574	.003
	hitfollow	558,100	1,172,248	.003
	bigfolo	574,823	1,157,672	.003

ity properties. As with egonets, we again see a stark contrast in the boomerang structure of these two providers. Figure 8.3a shows the dense internal connectivity of bigfolo's fake followers (as we saw in Figure 8.2h), in conjunction with the sparser and less compact external connectivity to friends. Conversely, Figure 8.2d shows sparse internal connectivity between twitterboost's fake followers on the left, but dense near-bipartite external connectivity to the customers (including honeypots) on the right.

Table 8.3 (top) gives summary statistics about premium boomerang networks, which substantiate the following:

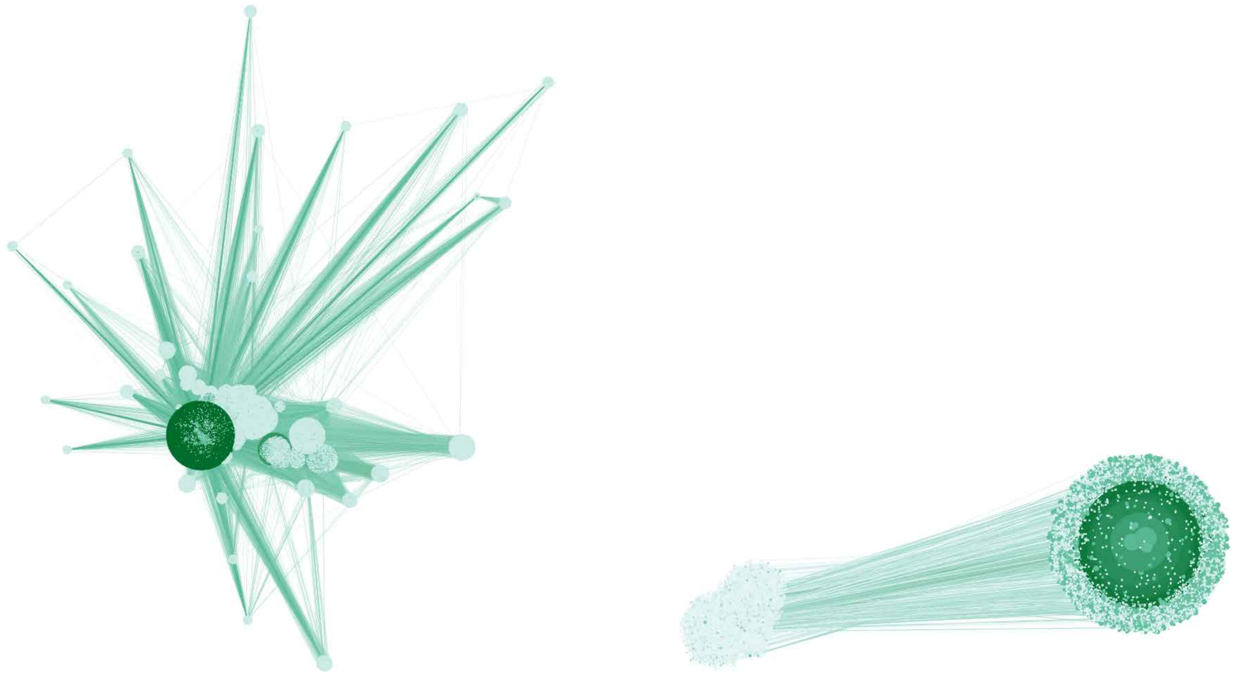
Insight 2 (Boomerang Density)**.** *Premium fake followers are frequently reused to follow customers, resulting in dense external connectivity in the boomerang network. Freemium fake followers are less reused to follow customers, and hence have sparse external connectivity.*

Interestingly, we see that the relative values of these statistics are inverted for the boomerang networks from the egonets – unlike for egonets where the density metric was an order of magnitude higher for freemium providers, the bipartite density in boomerang networks is instead an order of magnitude higher for the premium providers.Note that the premium providers' bipartite density indicates that nearly 1-2% (a huge amount) of all possible edges between the fake followers and their combined set of friends exists. The node to edge ratios are also much higher for premium providers – fastfollowerz and intertwitter are 1:14, and devumi and twitterboost are roughly 1:37 compared to only 1:2 for the freemium providers.

The freemium boomerang network statistics in Table 8.3 (bottom) again establishes the second part of the insight. This is further substantiated by the observation that freemium providers have an order of magnitude lower bipartite density than premium ones. We also observe that freemium boomerang networks have higher number of nodes than the premium counterpart. This is intuitive as freemium followers are otherwise genuine accounts, they have an expansive set of true friends, whereas premium fake followers are all synthetic accounts.

Network Overlap Patterns

In our analysis thus far, we noticed that various providers have different levels of evident overlap in the fake followers they deliver between their 2 honeypots. How extensive is this overlap? Do these providers reuse accounts in the same ways? Furthermore, is there any overlap between the



(a) bigfolo (fre.)

(b) twitterboost (pre.)

Figure 8.3: **Freemium followers have dense internal and sparse external connectivity (top), and vice versa for premium followers (bottom).** Subplots show boomerang networks, with darker node color and larger size indicating higher in-degree.

Table 8.4: Fraud providers have varying account reuse habits.

	Service	# Nodes	Overlap	Est. Pool # Nodes
Premium	fastfollowerz	1,064	.996	1,064
	intertwitter	1,049	.953	1,051
	devumi	2,679	.024	55,719
	twitterboost	26,78	.024	55,677
Freemium	plusfollower	918	.815	954
	newfollow	753	.765	798
	hitfollow	780	.802	814
	bigfolo	747	.774	791

followers across providers? Here, we shed light on these questions.

Intra-Network Patterns First, we study *intra-network overlap*, describing overlap between the fake follower nodes within each service. Table 8.4 shows the overlap coefficients between the honeypot followers for each service. Assuming the followers for each honeypot are randomly sampled from the service's account pool, we additionally compute the estimated total number of fake accounts currently in the fraud provider's hands using MSE.

The various degrees of overlap and commensurate estimates of pool size suggest the following insight:

Insight 3 (Varying Delivery Structure)**.** *Service providers have varying methods for account reuse in efforts to to distribute suspicion across their account pools.*

We observe that the freemium providers tend to have a high, 0.8 overlap which results in an estimated pool size slightly larger than either of the two sets of honeypot followers. However, the premium providers have an interesting split which reveals that `fastfollowerz` and `intertwitter` have very high, near 1.0 overlap, resulting in the pool size being roughly equal to each set of followers. This indicates that the pool is reused almost exactly for multiple customers. Conversely, `devumi` and `twitterboost` have near 0 overlap. As a result, we estimate that the pool size is quite large, containing over 55K total fake accounts.

While we cannot be certain without further investigation, these providers likely have different means of selecting and shifting the pool of active fake followers. For example, the pools used in `fastfollowerz` and `intertwitter` may cycle between a number of different "sub-pools" based on time, customer account features, or random choice. Conversely, the evidently much larger estimated pool size for `devumi` and `twitterboost` suggests that they may each have a single, large fixed pool of usable accounts from which followers are sampled regardless of other factors.

Inter-Network Patterns Thus far, we have established that providers reuse multiple follower accounts across customers in order to turn a better profit. But how far does this reuse go? Are any accounts responsible for delivering fake links to customers from different providers? To answer these questions, we study the pairwise *inter-network overlap* of followers between providers.

Table 8.5 shows an 8×8 matrix with the pairwise overlap coefficients. Given the number of nonzero entries, we draw the following surprising insight:

Insight 4 (Collusion)**.** *Service providers seem to collaborate with and draw from each other to commit fraudulent actions.*

We notice that there is substantial overlap within the freemium and premium providers. While `fastfollowerz` and `intertwitter` share no accounts with the other premium providers, `devumi` and `twitterboost` have a .07 overlap. Comparatively, all 4 freemium providers have a large 0.6-0.7 overlap, indicating that most of their fake accounts are *the same*. Furthermore, the set of followers for freemium and premium providers have 0 overlap, substantiating that followers in freemium providers are otherwise real accounts whereas those in premium providers are synthetic.

Nonzero overlap between providers is an interesting finding – it is indicative of either a willingness to share follower accounts between fraud providers, or commonality in leaked or hijacked accounts. Upon further inspection, we notice a number of suggestive findings:

- Overlapping providers shared domain WHOIS protectors.
- Overlapping premium providers use the same Yoast SEO plugin and stylesheets.
- All freemium providers have two-column sites, advertised up to 30K followers, and priced from £9.99.
- All fremium providers contained the line: *"[service] is Not Affiliated With OR Endorsed By Twitter.com."*

Table 8.5: Fraud providers share follower accounts.

		fastfollowerz	intertwitter	devumi	twitterboost	plusfollower	newfollow	hitfollow	bigfolo
Premium	fastfollowerz	1.0	0	0	0	0	0	0	0
	intertwitter	0	1.0	0	0	0	0	0	0
	devumi	0	0	1.0	.07	0	0	0	0
	twitterboost	0	0	.07	1.0	0	0	0	0
Freemium	plusfollower	0	0	0	0	1.0	.65	.69	.64
	newfollow	0	0	0	0	.65	1.0	.64	.63
	hitfollow	0	0	0	0	.69	.64	1.0	.63
	bigfolo	0	0	0	0	.64	.64	.63	1.0

Table 8.6: Per-service entropy (in bits) over account attribute distributions.

	Service	**Created (year)**	**Def. Prof.**	**Def. Prof. Image**	**# Favorites**	**# Followers**	**# Friends**	**# Lists**	**# Statuses**	**Geolocation**	**Lang.**	**Protected**	**UTC**	**Verified**
Premium	fastfollowerz	1.37	.63	.01	3.65	2.73	2.73	2.99	3.8	.00	.06	.00	1.04	.00
	intertwitter	2.99	.82	.94	4.04	3.54	2.63	2.53	4.31	.67	2.55	.56	1.97	.18
	devumi	1.13	.97	.02	1.05	1.54	1.17	2.49	1.18	.00	.00	.00	1.42	.00
	twitterboost	1.13	.97	.03	1.05	1.56	1.16	2.51	1.15	.00	.00	.00	1.41	.00
Freemium	plusfollower	1.82	.93	.73	4.18	3.76	3.38	2.73	4.40	.54	2.04	.30	1.70	.00
	newfollow	1.68	.90	.75	4.20	3.70	3.32	2.64	4.37	.55	1.99	.28	1.62	.00
	hitfollow	1.78	.93	.73	4.14	3.76	3.32	2.72	4.37	.52	2.01	.30	1.70	.00
	bigfolo	1.88	.92	.75	4.20	3.74	3.34	2.72	4.40	.56	2.05	.32	1.71	.00
	Max Entropy:	3.46	1.00	1.00	5.00	5.00	5.00	5.00	5.00	1.00	5.13	1.00	5.29	1.00

8.2.3 Attribute Observations

In this section, we study the similarities and differences in account attributes of fake followers. Table 8.6 shows per-service, per-attribute entropy in bits for a variety of user attributes. The account attributes include creation year, default profile and profile image booleans, favorites count, followers count, friends count, lists count, statuses count, geolocation enabled boolean, language identifier, protected statuses boolean, UTC timezone, and a Twitter verification boolean which corresponds to high-profile, "famous" accounts. These attributes have varying outcome spaces. Creation date has 11 possible years (2006-2016), since Twitter was founded in 2006. Booleans have 2 possible outcomes (T,F). We encountered 35 different language identifiers and 39 UTC timezone settings. For count features, we logarithmically discretized the space into 32 bins from 1 to 1M to capture the wide range of activity levels. For each service, we aggregate attribute values and compute the entropy over the outcomes. The table shows the actual sample entropy in addition to the maximum possible (uniform) entropy. As previously mentioned, lower entropy indicates high synchronicity between followers. Note that a difference in entropy of 1 bit corresponds to twice the predictability.

The most striking insight from Table 8.6 is as follows:

Insight 5 (Entropy Gap)**.** *Premium service providers deliver followers with low entropy, high* *regularity attributes, whereas freemium service providers have more attribute disparity.*

We notice that the premium providers have substantially lower entropy values in many attributes versus freemium providers, and even near 0 entropy in other attributes like geolocation. We elaborate on the specific differences next.

Account Creation

devumi, twitterboost and fastfollowerz have very low creation year entropy compared to freemium providers. While both freemium and premium accounts tend to be created more recently (perhaps because of higher suspension rate in older accounts), premium providers have a heavy bias towards recently created accounts (>2014).

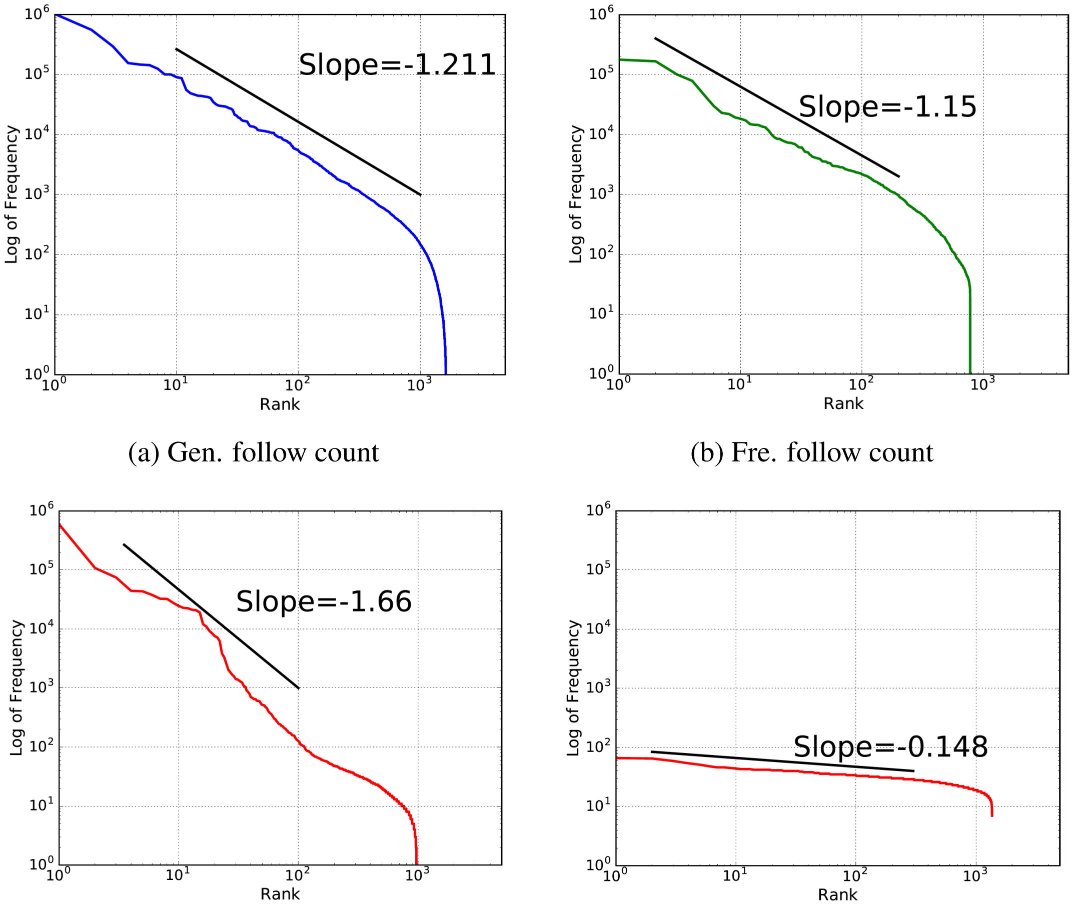
Profile Defaults

fastfollowerz has a much lower entropy than other providers in terms of default profile – we found that >84% of these accounts did not have a default profile, whereas default profiles are actually *more common* than not in freemium accounts. Surprisingly, fastfollowerz, devumi and twitterboost also have near 0 entropy for profile image compared to the much higher entropy for freemium providers. We find that premium followers almost always set a custom image, suggesting that the information was fabricated or stolen from real users. Conversely, default profile images are common for freemium service accounts – this is intuitive, most real users do not fully customize their profiles.

Action Counts

devumi and twitterboost have much lower entropy for action counts (favorites, followers, friends, lists and statuses) compared to freemium providers. fastfollowerz also exhibits lower entropy. As Figure 8.1d shows, there is even more variation between premium providers. Figure 8.1d shows that intertwitter (P1 “smart”) follower counts are disparate and closer to genuine users’ entropy, unlike other premium fraudsters (P2 “naïve”) who behave robotically. Comparatively, freemium followers have lower follower count entropy compared to genuine ones, which is intuitive as while the freemium follows are real accounts, their follower counts are not independent from each other due to the follows traded between themselves. Figure 8.4 shows the rank-frequency plots for follower counts for various follower types. The plots substantiate our observations on entropy, and also show that different user types exhibit differences with regards to power-law fit, which is expected for skewed distributions on social networks. While entropy values in this paper are computed empirically using the samples from Table 8.2, accounts on real networks have varying follower counts, leading to different entropy estimates even when drawn from the same distribution.

We noticed similar patterns in entropy for status and favorite counts as well. The lower entropy of action counts characteristic of premium providers stems from the variety of options premium providers have for Twitter engagement – in addition to fake followers, the premium providers also offer fake retweets and favorites services. Thus, premium providers are incentivized to reuse accounts for multiple types of fraud, and when done naïvely result in high synchrony in “serviceable” attributes.



(a) Gen. follow count

(b) Fre. follow count

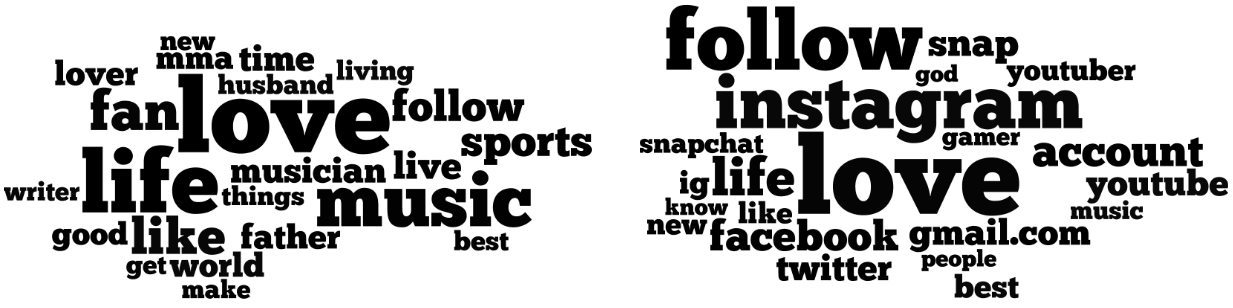
(c) P1 (“smart”) follow count

(d) P2 (“naïve”) follow count

Figure 8.4: **Rank-frequency plots reveal different patterns in follower counts of various follower types.** Note that genuine follower counts in (a) reflect traditional power-law behavior with a common exponent (~ 1.2) and are linear in log-log scale. Freemium counts in (b) fit similarly, despite with a slightly lower exponent (~ 1.15). Comparatively, “smart” premium counts in (c) fit a power law but with much higher exponents (~ 1.66). Interestingly, we find that “naïve” premium followers do fit a power law, but have unnaturally low exponents ($\sim .148$) due to their low entropy and highly concentrated, robotic behavior.

User Settings

fastfollowerz, devumi and twitterboost all have near 0 geolocation, language, and tweet protection entropy. Of these, all devumi and twitterboost accounts use the US English language setting, have geolocation disabled and do not protect tweets. fastfollowerz has a slightly higher language entropy of .06, but we found that all fastfollowerz accounts were either using US or GB English, suggesting a heavy premium bias for English accounts. We also found that premium followers almost entirely have USA timezones. “Smart” intertwitter followers’ high language entropy from Figure 8.1d suggests an aim to better camouflage user attributes compard to the



(a) Premium

(b) Freemium

Figure 8.5: **Freemium followers have social media (Facebook, Instagram, Snapchat) focused descriptions (right), whereas premium followers have wordy descriptions (left).**

"naïve" providers. Given that `intertwitter` also has some verified accounts, we hypothesize that the accounts may be hijacked ones. This is in contrast with freemium providers, which have much higher frequency of enabled geolocation, variance in language and protected tweets. Figure 8.1d also shows that freemium followers tend to appear similar to genuine ones as they are otherwise real user accounts. However, we find that freemium followers have higher language entropy than genuine ones, as freemium followers are spread over many languages whereas genuine followers tend to disproportionately speak their followee's language (i.e. if a user speaks Spanish, most of his followers speak Spanish).

Furthermore, all 4 freemium providers and `twitterboost/devumi` have extremely similar attribute entropy over their fake followers respectively, further substantiating Insight 4.

In addition to the attributes reported in Table 8.6, we also studied the 160-character user description field. The description field essentially contains the high-level summary of what the user aims to appear as to other Twitter users, and is thus interesting to analyze. We ask: what, if any, are the differences between freemium and premium follower descriptions?

Figure 8.5 shows two wordclouds, aggregated over description text across all premium and freemium followers respectively. Font size corresponds to relative frequency in the text. For clarity, we remove common stopwords. We arrive at the following insight:

Insight 6 (Clout vs. About)**.** *Freemium followers tend to have descriptions focusing on social media clout, whereas premium followers tend to talk about themselves.*

Figure 8.5a (premium), has words like "musician," "lover," "writer" and "sports", corresponding to descriptive personal details – these are likely copied from genuine users. Conversely, Figure 8.5b (freemium) has terms like "snapchat," "youtube," and "instagram", as these users try to increase clout by advertising their other, real social media pages, i.e., "*follow me on snapchat*."

8.3 Assessing Discriminative Power of Entropy Features

Thus far, we have highlighted a number of distributional differences between fraudulent and genuine users. Can we leverage these differences to discriminate user behaviors? In this section, we evaluate a number of attribute features on their discriminative power in a supervised setting.

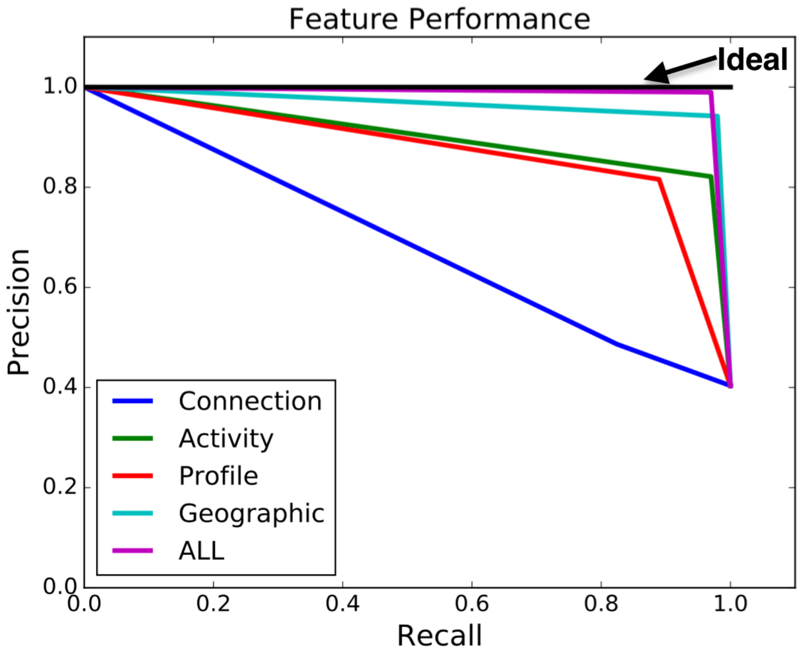


Figure 8.6: **Leveraging all features together gives the best detection performance.**

We classified the engineered entropy features from Table 8.6 into the following groups based on feature type:

- *Connection*: # Followers, # Friends
- *Activity*: # Statuses, # Lists, # Favorites
- *Profile*: Default Profile (and Image), Verified, Created
- *Geography*: Language, UTC
- *All*: the union of all above features

Note that while we nominally refer to these features as above, they refer to the *entropy of the feature over account followers*, rather than raw values of the account itself.

We evaluate these features using binary classification (genuine vs. fraudulent) as is traditionally done in practice. We use a Support Vector Machine (SVM) with radial basis function (RBF) kernel and 10-fold cross validation as the classifier of choice, but any out-of-box classification method could be used. Our carefully assembled ground-truth dataset consists of 307 fraudulent users and 200 genuine users, whose features are computed over their followers. The fraudulent accounts are a combination of premium and freemium honeypots as well as accounts whose profiles have been listed on freemium providers' websites as users of the service. We define our fraudulent set over this multitude of account types with various properties in order to demonstrate generality. The genuine accounts belong to well-known academics in machine learning and data mining. We avoid using randomly sampled Twitter users, as previous works have shown a non-trivial amount of fake accounts on Twitter which may excessively corrupt our ground-truth genuine set. In practice, getting additional ground-truth labels is a very costly endeavor and requires careful manual inspection for each individual case.

Figure 8.6 shows the relative performance of our feature groups in terms of overall precision and recall. We notice that *Connection* features perform comparatively poorly, *Profile* and *Activity* features perform better, *Geography* performs even better, and the combination *All* performs near-ideal with .98 precision and .95 recall (much higher recall than supervised approaches which use raw account features for Twitter spam classification [195]). Thus, we conclude that our proposed entropy features are highly reliable in discerning genuine from fraudulent users. The added benefit of using the entropy-based features is that it is much harder to control for from the fraudster's perspective – this is because while the fraudster has significant control over his own account's properties, he has limited ability to influence who follows him.

8.4 Discussion

The analysis in this work has a number of important implications on fraud detection in practice. We detail these below.

Multimodal Detection: Using individual signatures to find one type of fraud tends to be at the expense of finding other types. For example, clique detection primarily focuses on freemium fraud, whereas bipartite core detection focuses on premium fraud. Using complementary methods is a promising strategy.

Importance of Time: Varying account reuse policies makes temporal granularity an important consideration in graph-based fraud detection. While analysis on a low granularity graph can reveal dense fraudulent structure in frequent reuse regimes, it may never do so for low reuse regimes. Higher granularity can be useful in these cases.

Deceptive Account Attributes: Using individual account attributes to label fraudsters is of limited use. Our work suggests that most freemium fraudsters are actually real users with real profile attributes – they may be resistant to such detection schemes. Conversely, leveraging an account's follower's attributes shows promise in bridging this gap.

Total vs. Partial Fraud: Different types of fraud may call for different penalties. While the implication "has one fake link → has all fake links" seems true for premium fraudsters, it is not for freemium ones. Removing fake links vs. suspending fake accounts is a promising way to penalize such fraudsters and minimize false positives.

The need for multimodal anti-fraud mechanisms suggests a shift in the detection paradigm from drawing a two-class boundary between genuine and "one-hat-fits-all" fraudulent users, to a more complex multiclass boundary between genuine, premium fraudulent, freemium fraudulent, and other fraud types which may be discovered in the future.

8.5 Conclusion

In this work, we aimed to study the nature of modern link fraud regimes. To this end, we setup honeypot accounts on Twitter, purchased fake followers for them from a variety of fraud-provoding services, and carefully instrumented a data scraping process to capture their behaviors. Specifically, we studied the local network connectivity of fake followers via the egonet and proposed boomerang networks, as well as attribute distributions over profile features and account actions. Our analyses showed that there are multiple types of link fraud (we discover at least two: *freemium* and *premium*) with varying behaviors regarding internal and external network connectivity, disparity in attribute homogeneity across followers, and differences in descriptive word-usage in Twitter bios. Furthermore, we found fascinating evidence that service providers have varying types of account-reuse policies and seem to collude with each other on a number of fronts. Furthermore, we proposed the use of first-order entropy features taken across account followers' attributes to discern fraudulent from genuine accounts, and showed that these features were able to attain near-perfect F1 score on our ground-truth dataset. Holistically, our work offers several implications for practical fraud detection including multimodality of fraud behaviors, the importance of temporally sensitive algorithms, usefulness of first-order versus zeroth-order features, and disadvantages of account-based versus link-based fraud targeting.

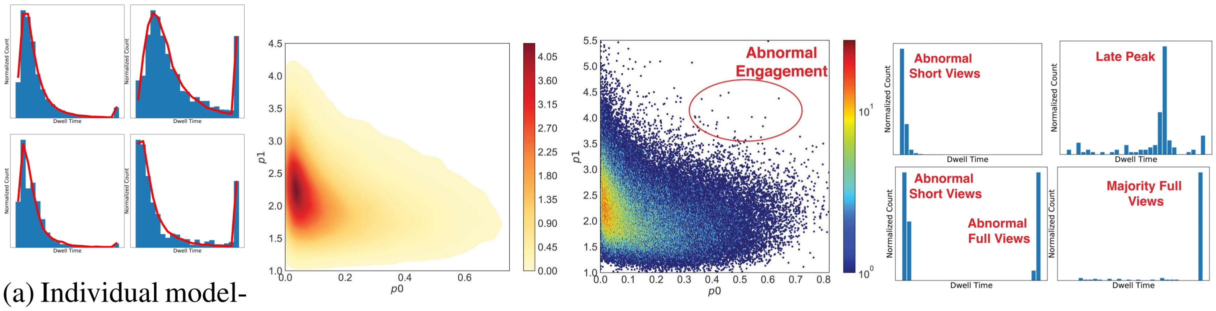
Hemank Lamba, Neil Shah. ”Modeling dwell-time engagement on visual multimedia”. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2019.

CHAPTER 9

MODELING DWELL TIME ENGAGEMENT FRAUD

Visual multimedia is one of the most prevalent sources of modern online content and engagement. However, despite its prevalence, little is known about user engagement with such content. For instance, how can we model engagement for a specific content or viewer sample, and across multiple samples? Can we model and discover patterns in these interactions, and detect outlying behaviors corresponding to abnormal engagement? In this paper, we study these questions in depth. Understanding these questions has implications in user modeling and understanding, ranking, trust and safety and more. For analysis, we consider content and viewer *dwell time* (engagement duration) behaviors with images and videos on Snapchat Stories, one of the largest multimedia-driven social sharing services. To our knowledge, we are the *first* to model and analyze dwell time behaviors on such media. Specifically, our contributions include (a) *individual modeling:* we propose and evaluate the UM-DP, LM-DP and V-DP parametric models to describe dwell times of unlooped/looped media and viewers which outperform alternatives, (b) *aggregate modeling:* we show how to flexibly summarize the respective joint distributions of multivariate parametrized fits across many samples using Vine Copulas in the analog UM-AM, LM-AM and V-AM models, which enable inferences regarding aggregate behavioral patterns, and offer the ability to simulate real-looking engagement data (c) *anomaly detection:* we demonstrate our aggregate models can robustly detect anomalies present during training (0.9+ AUROC across most attack models), and also enable discovery of real dwell time anomalies.

The recent years have brought about a tremendous increase in proliferation of visual multimedia content in the form of images and videos. Internet users watch *1 billion* hours of YouTube video [29], share more than *95 million* images and videos on Instagram [281], and spend an average of *30 minutes* on Snapchat every day [282]. *Dwell time*, or engagement duration, is one of the key means of implicitly describing user interactions with content. In contrast to explicit features such as likes and follows, dwell time is not afflicted by low response rates and reporting bias. Content with high dwell time is considered more interesting and valuable to viewers, and



(a) Individual modeling (b) Group modeling (c) Anomaly detection (d) Dwell time anomalies

Figure 9.1: Our work discusses (a) state-of-the-art parametric models for individual sample dwell times which closely mirror empirical data, (b) flexible copula modeling of aggregated multivariate parameter fits, (c) utilization of aggregate models for detecting dwell time engagement anomalies which (d) reflect abnormal behaviors radically inconsistent with most samples.

indicate user attentiveness and satisfaction. Dwell time has thus been used as a central feature in content recommendation [141, 307, 308]. However, despite its value, prior work has left a considerable gap in modeling and analysis of dwell times on visual multimedia.

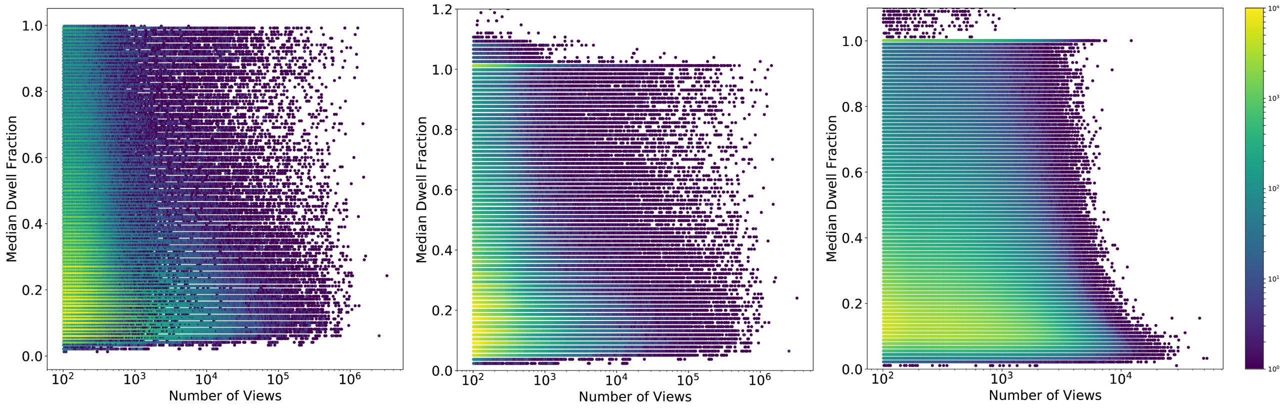
With the insight that dwell time can influence recommendations, numerous online marketplaces have spawned , offering customers ways to increase perceived engagement via paid inauthentic "views"; searches for "buy Youtube views" or "buy Instagram views" show numerous services offering bundles of 1 thousand views for as little as 10¢. Such inauthentic engagement can disrupt recommendation algorithms, hurt advertiser profits, and increase user exposure to bad content. Despite this, prior work towards detecting abnormal viewer engagement using dwell times is nearly non-existent.

To bridge these gaps in behavior modeling and anomaly detection literature, we pose the following research questions:

- **RQ1. Individual Modeling**: How can we describe the dwell time distribution for a given content/viewer sample?
- **RQ2. Aggregate Modeling**: How can we jointly model dwell times across many content/viewer samples?
- **RQ3. Anomaly Detection**: Can such models help us detect dwell time engagement anomalies?

Dwell times have primarily been studied in the context of documents like webpages [141, 305] and short articles [308]. To the best of our knowledge, ours is the first work that tackles the problem of modeling dwell times on *visual multimedia*. Our context poses a number of non-trivial challenges, including variety in varying content durations, media formats (looped and unlooped content) and behavioral diversity. Moreover, the sheer scale of engagement data is huge, necessitating scalable solutions for modeling. Our approach posits three core contributions, mirroring RQ1-RQ3:

- **C1. Individual Modeling:** We propose concise, interpretable parametric models which match empirical dwell time behaviors. Our proposed Um-Dp (see Figure 9.1a), Lm-Dp and V-Dp models for characterizing looped/unlooped media dwell times consistently outperform alternatives in terms of goodness-of-fit via 2-sample test and log-likelihood.



(a) Unlooped Media (b) Looped Media (c) Viewers

Figure 9.2: Median dwell time ratios vs. number of views on (a) unlooped and (b) looped media, and (c) viewers show outliers which exhibit excessively high dwell times compared to normal engagement patterns of similar view-count peers.

- **C2. Aggregate Modeling:** We propose aggregate models for looped/unlooped media (LM-AM/UM-AM) and viewer (V-AM) dwell times, which utilize copulas to preserve multivariate dependency structures and model joint distributions of individual parameter fits (see Figure 9.1b). These models parametrically approximate original data with constant space, offer scalable inference, are temporally consistent and are also generative.
- **C3. Anomaly Detection:** We demonstrate that our aggregate models can be used to easily discover those with abnormal engagement (see Figures 9.1c/d). Experiments show our approach enables robust anomaly detection against simulated attacks (0.9+ AUROC in most experiments), and detects anomalous dwell time engagement behaviors on real data.

Though our work uses viewing data from Snapchat, we expect that given the diversity and scale of viewers and media settings that we consider, our findings should generalize on other visual multimedia platforms which support similar visual content types.

9.1 Related Work

We discuss prior work in (a) temporal behavior modeling, and (b) detecting anomalous viewership.

Temporal behavior modeling. Prior work in dwell time modeling primarily focuses on recommendation and prediction of webpages and text documents. [308] explores interpreting dwell times as "pseudo-votes" for content recommendation of short-text documents; using a Log-normal distribution to model dwell times. [305] discusses using dwell times for re-ranking webpage results. [178] demonstrates that webpage features predicts the Weibull distribution modeled dwell times of webpage visits. [47, 141] also discusses predicting dwell times on webpages and YouTube videos, respectively. Additionally, [30] proposed using gamma, weibull and exponential distribution to model dwell times. Several works focus on modeling temporal behaviors other than dwell times. [134] and [67] propose using the Log-logistic distribution to describe user interarrival times between search queries and forum comments. [287] uses a left-truncated Log-logistic model to describe human phone-call durations. Significant amount of work has been done in anomaly detection for time-series [108], however our work is not concerned with modeling sequences, bur instead underlying distribution of dwell times.

Overall, unlike ours, none of the prior works (a) involve parametric dwell time modeling, (b) tackle general visual multimedia, and (c) model both users and content.

Detecting anomalous viewership. Prior literature in detecting anomalous viewership is sparse. [190] analyzed the fake view detection capacities of several video-sharing services including YouTube, DailyMotion, and Vimeo under synthetic attack models, demonstrating that all services were susceptible to simple attacks of fixed interarrival time views across IP addresses. [47] propose using user, IP and video entropies in a supervised model to detect abnormal engagement; however, their approach requires intensive manual labeling.[252] proposes using temporal view features in a livestreaming setting to detect distributional anomalies, but is non-parametric and does not expressly model dwell time behaviors, while being undefined for viewer and content anomalies.

Overall, unlike ours, none of the prior works (a) utilize implicit dwell time rather than explicit feedback, (b) tackle general visual multimedia, and (c) are unsupervised.

9.2 Data Description

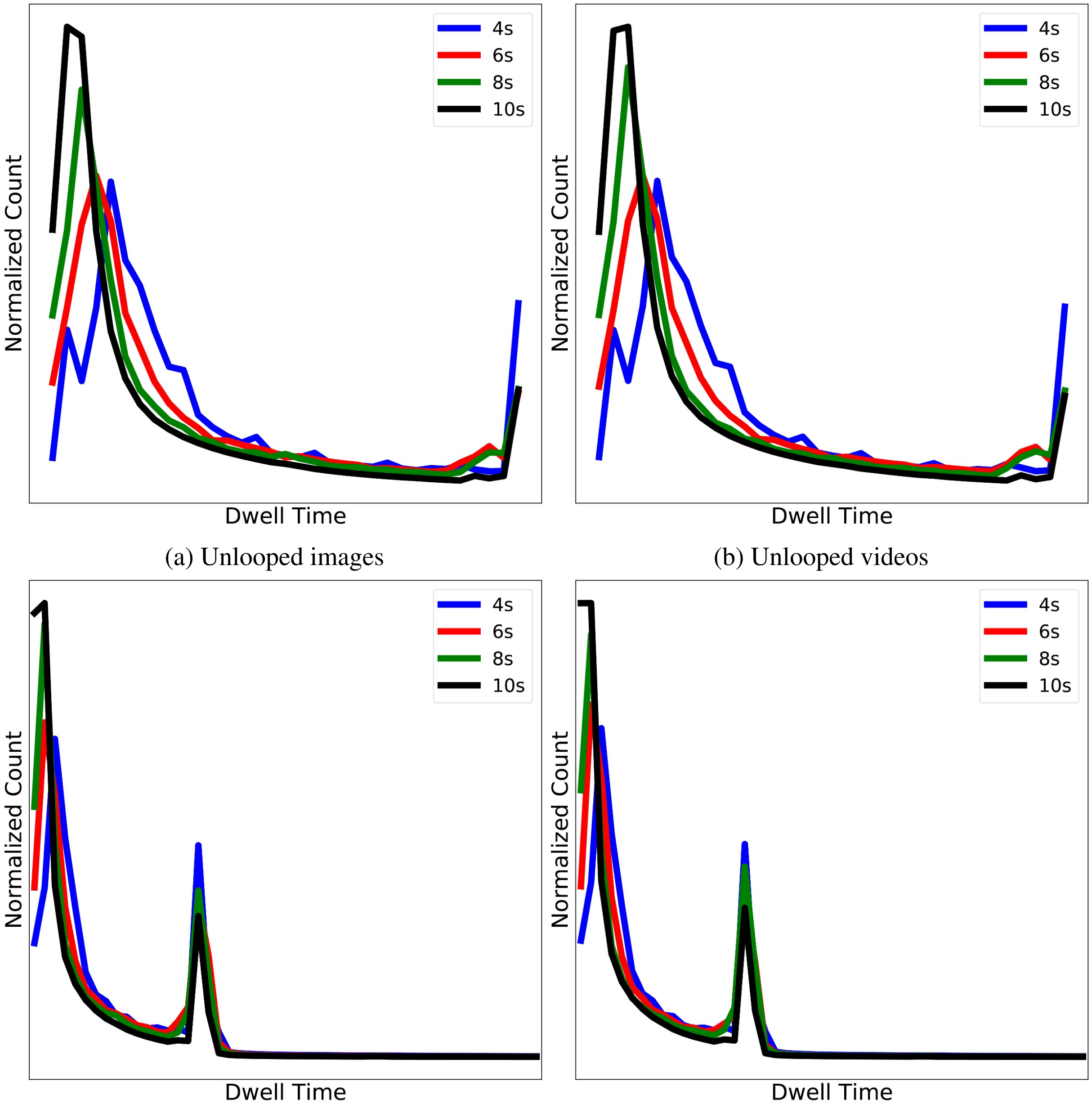
Table 9.1: Dataset summary

Unique media samples	300 thousand
Images	208 thousand
Videos	92 thousand
Unlooped	102 thousand
Looped	198 thousand
Unique viewers	24 million
Total views	273 million

In this work, we study an industrial-scale media engagement dataset from Snapchat, one of the largest social multimedia-driven content sharing services. Snapchat enables users to share visual multimedia content to their “My Story,” which can be optionally exposed to the entire userbase. Specifically, users can share ephemeral (purged in 24 hours) content (images or videos) with duration of up to 10 seconds, and adjust loop settings to unlooped (views automatically terminate upon completion) or looped (repeat indefinitely).

Our dataset consists of engagement associated with a large set of publicly posted “My Story” contents, and of the associated viewers for a long-enough time period sufficiently accounting for complete 24-hour observation of engagement with all samples[1]. Table 9.1 details several

[1] Due to privacy reasons, we obscure certain sensitive details (timeframes and certain axes values) while communicating our insights.



(a) Unlooped images

(b) Unlooped videos

(c) Looped images

(d) Looped videos

Figure 9.3: Aggregated dwell time ratio statistics for varying media types and durations inform our modeling choices: treat images and videos similarly, and unlooped and looped content distinctly.

key summary statistics of our dataset. All content samples and viewers have 100+ associated views/data-points, enabling us to draw reasonably reliable inferences about engagement.

9.3 Initial Observations

Before delving into details, we conduct several exploratory analyses to motivate and direct our approach and give intuition for our subsequent modeling choices. Firstly, we aim to understand dwell time behavior across the entire dataset, to determine patterns and anomalies in dwell times across different content and viewers.

Since different content samples have varying durations, we normalize all dwell times with respect to these in order to compare them. Henceforth, when we mention "dwell time," we consider instead the dwell time *fraction* or *ratio*. Thus, dwell time ratios of views on unlooped media must lie in $(0, 1]$, whereas dwell time ratios on looped content can lie on $(0, \infty)$.

Figure 9.2 shows quantized heatmaps of median dwell ratio of unlooped/looped media (9.2(a) and 9.2(b), respectively) and viewers (9.2c) versus view count, with brighter colors indicating logarithmically increasing density and darker colors denoting sparsity.Intuitively, sparsity increases towards the right of each plot due to skewed view count distributions, and towards the top of each plot, as few entities have high dwell ratios. Additionally, there are sparse entities in all plots which have very low dwell ratios. In all cases, we observe well-defined regions of high density.This suggests the following key observation, which motivates our modeling and anomaly detection goals.

Key Observation 1 ((In)Consistencies in Visual Multimedia Dwell Times)**.** *There exist patterns and anomalies in content and viewer dwell time engagement on visual multimedia.*

Next, we consider collective differences between unlooped and looped media, and their implications for dwell time modeling. Figure 9.3 shows the collective dwell ratios across our entire dataset, for unlooped images and videos in 9.3(a-b), and their looped counterparts in 9.3(c-d). The stark differences in distribution shape is apparent; unlooped dwell ratios are effectively censored at 1.0, where they achieve a second peak after a tapered drop. However, while looped dwell ratios exhibit a similar decay and noticeable peak at the first view "completion," (near mid-plot) they show a decreasing but nonzero probability afterwards due to differences in feasible view duration across the media types. This suggests the following:

Observation 1 (Looped/Unlooped Media Dwell Time Disparity)**.** *Looped and unlooped media require characteristically different dwell time models, due to the differences in support over dwell time ratios of* $(0, 1]$ *and* $(0, \infty)$*, respectively.*

Lastly, we consider the effect of different media type (image and video) on dwell ratios. By comparing Figures 9.3(a)/(c) with 9.3(b)/(d), we can observe that images and videos actually admit very similar dwell times. Despite videos being intuitively "richer" than images, the plots mirror each other. Moreover, since we observe no significant differences in the "stickiness" across the collective media type splits, we hypothesize that a significant portion of users' decision to engage with content may actually occur *before* the user accesses the content, for example due to self-selection and preferences towards certain content. Our major takeaway regarding media types is thus

Observation 2 (Image/Video Dwell Time Parity)**.** *Dwell time similarities across image and video engagement suggest that they can be modeled characteristically similarly.*

Given these observations, we next discuss our proposed parametric models for dwell time distributions of individual content samples and viewers; parametric models are appealing due to their conciseness and interpretability over nonparametric alternatives.

9.4 Individual Dwell Time Modeling

How can we parametrically model the dwell time distributions of multimedia content and viewers? In this section, we first propose "dwell processes" to generatively model the multimedia content for both looped and unlooped media. Following this, we posit the same contributions for viewers. In both cases, we give the intuition behind our modeling approaches, discuss efficient parameter inference procedures and validate against alternatives using goodness-of-fit metrics.

9.4.1 Multimedia Content Modeling

Looped Content

We begin by discussing modeling of looped content. Views on such content are unbounded, and dwell time ratios can range from $(0, \infty)$. Given our earlier insights regarding long-tailed dwell times from collective analysis in Figure 9.3, we consider several suitable distributions that may be able to model such shapes. In our preliminary analyses, we observed that the tails of many samples matched quite closely with Log-logistic distribution, defined as

Definition 1 (Log-logistic (LL) Distribution)**.** *Let T be a non-negative continuous random variable, such that $T \sim LL(\alpha, \beta)$. The PDF and CDF of T are given by*

$$f_{LL}(t; \alpha, \beta) = \frac{(\beta/\alpha)(t/\alpha)^{\beta-1}}{(1+(t/\alpha)^{\beta})^2} \qquad F_{LL}(t; \alpha, \beta) = \frac{1}{1+(t/\alpha)^{-\beta}}$$

where $t \in [0, \infty)$, and α(scale), $\beta > 0$(shape) are the parameters.

Note that the LL distribution admits the same support as our use-case for looped content, but does not do so for unlooped content. We propose using the original, unmodified LL distribution as the core of our LM-DP (**L**ooped **M**edia **D**well **P**rocess), which can be written generatively as

Definition 2 (Looped Media Dwell Process (LM-DP))**.** *Sample each dwell time ratio $t_i \sim LL(\alpha, \beta)$.*

Use of LL distribution over alternatives is justified for several reasons. LL is widely used in survival modeling and has a hazard function implying that the longer a view has persisted, the longer it will continue to do so [21]. Also, it has demonstrated success in modeling other real-world temporal phenomena [67, 134] besides visual multimedia dwell times, and as we will show below, it outperforms other candidate distributions in this task.

Inference of LM-DP. Inference of α and β cannot be computed in closed form. As a result, we infer parameters using the Nelder-Mead simplex method [154], which maximizes likelihood via iterative approximations, while converging quickly and accurately.

Validation of LM-DP. We validate the model both qualitatively (visually) in terms of empirical versus simulated dwell time probabilities, and quantitatively via the Kolmogorov-Smirnov (KS) 2-sample test. In Figure 9.5, we illustrate the strong match in empirical dwell time distributions and our superimposed model fits across several looped media samples of varying exposure durations, viewer counts and dwell time behaviors. For brevity, we show results only on 6 users, but most others exhibited similar quality of fit. Observe that LM-DP is able to well-approximate the peak and decay corresponding to view drop-offs reasonably well despite differences in distribution shapes across the samples, thus suggesting the appropriateness of our modeling choice.

To analyze the goodness-of-fit quantitatively, we perform KS tests comparing dwell times that were (a) empirically observed, with (b) those simulated by LM-DP using parameters inferred from MLE for each content sample. We compared LM-DP with four other alternative distributions which have previously been used for dwell time modeling in other contexts. These are CL-LN (Log-normal) [308], CL-IG (Inverse Gaussian) [107], CL-WB (Weibull) [178] and CL-G (Gamma) [141]. Figure 9.4(a) shows the sorted p-values reported across KS tests over samples reflecting the rejection probability for the null hypothesis H_0 that the empirical data and our simulated data are drawn from the same distribution. Assuming H_0 is true, the p-values should be uniformly distributed, manifesting as the $45°$ line. We observe that our proposed LM-DP using LL performs the best, with the CL-LN model the next closest, CL-IG/CL-WB and CL-G demonstrating significantly worse performance. Figure 9.4(b) further shows the percentage of samples that were fitted "successfully" (KS $p < .05$) given their view counts. Again, we observe that LM-DP outperforms competitors, modeling the vast majority of samples successfully (over 90% for samples with ≈ 100 views). Note that since KS tests and p-values are highly sensitive given large sample sizes (i.e. H_0 would be rejected even for minute differences between empirical and simulated data), the percent of successful fits decreases in all cases with high view count; however, given the skewed distribution of view counts, high view count cases constitute only a small fraction of the population. Table 9.2(LM-DP) further demonstrates the aggregated percentage of samples for which LM-DP outperforms the alternatives, according to both KS test p-values and negative log-likelihood (NLL) of the fitted models; note that these differences are significant and persistent across hundreds of thousands of samples. Additionally, since all the

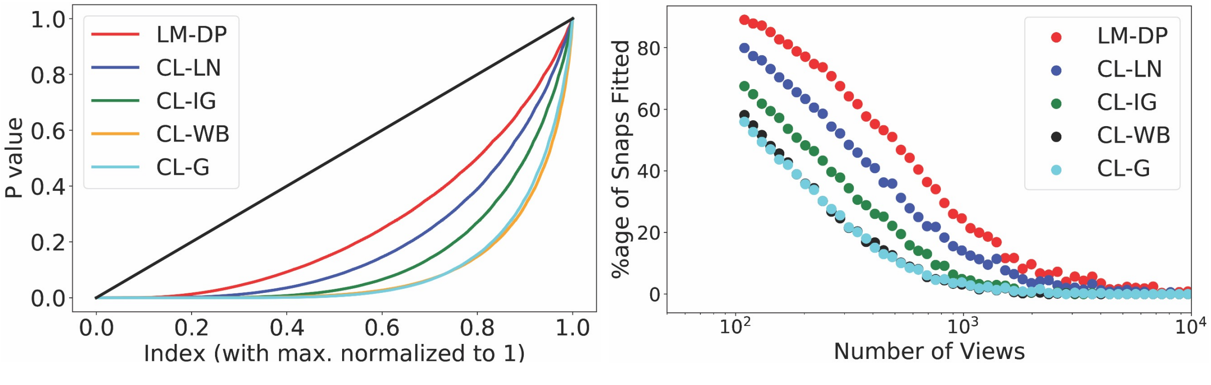


Figure 9.4: LM-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the $45°$ line, better the fit. (b) %age of samples where model fits were successful($p < .05$).

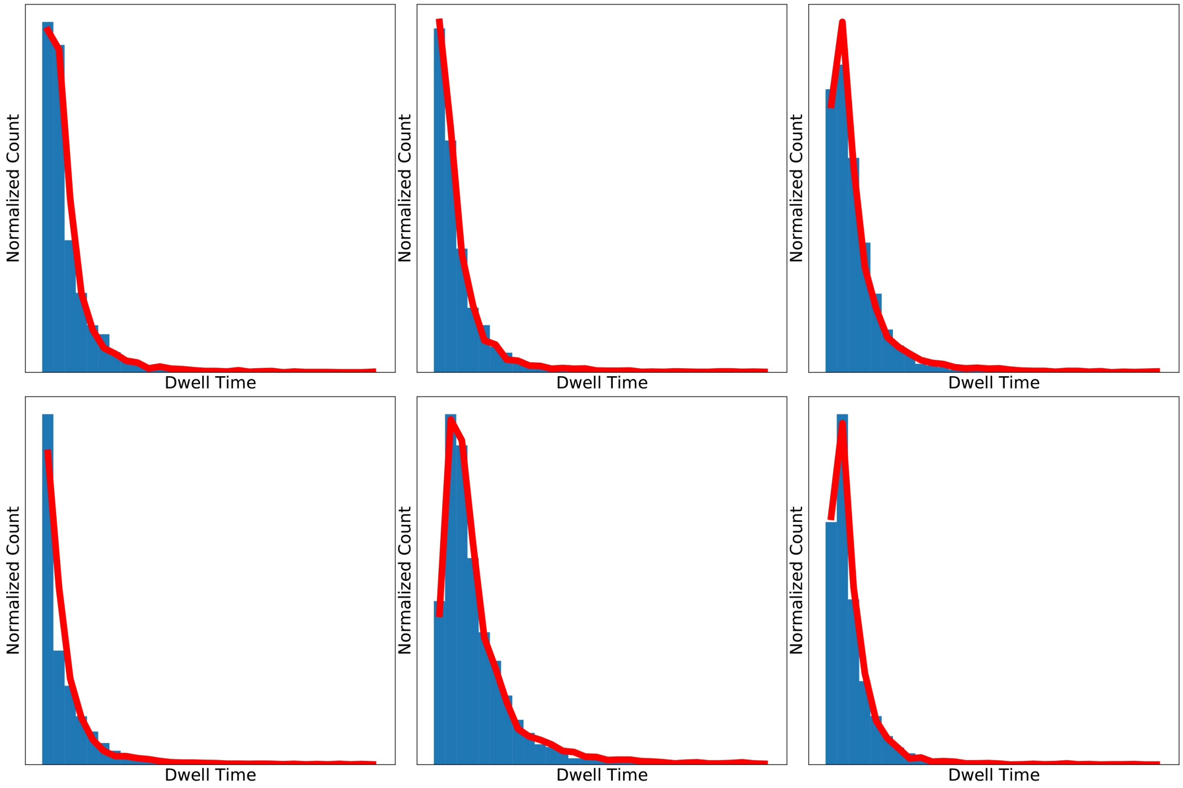


Figure 9.5: Proposed LM-DP (red) visually matches empirical dwell times (blue) across several looped media samples of varying patterns.

Table 9.2: % of instances where proposed models outperforms alternatives (higher is better, >50% implies superior performance).

LM-DP	**CL-LN**	**CL-IG**	**CL-WB**	**CL-G**
NLL	54.5%	82.4%	94.6%	93.7
KS	78.9%	86.2%	84.7%	86.7
UM-DP	**CU-LN**	**CU-IG**	**CU-WB**	**CU-G**
NLL	53.6%	78.2%	84.1%	85.5
KS	73.2%	86.7%	88.9%	90.2
V-DP	**CV-LL**	**CV-IG**	**CV-WB**	**CV-G**
NLL	93.6%	82.6%	99.1%	99.9%
K-S	52.8%	54.1%	81.1%	84.1%

models have same number of parameters, model complexity metrics are proportional to NLL and hence we do not explicitly mention them.

Unlooped Content

Unlike for looped media, unlooped views can have a maximum dwell ratio of 1.0 given viewing constraints (discussed in Observation 1). We observe that no typical continuous value distributions are able to handle this constraint on support natively. Therefore, we propose our UM-DP (**U**nlooped **M**edia **D**well **P**rocess) which significantly augments the LM-DP to handle this constraint. The model can be written generatively as

Definition 3 (Unlooped Media Dwell Process (UM-DP))**.** *Sample each dwell time ratio t_i as*

1. $c_i \sim Bernoulli(\theta)$
2. $t_i \sim \begin{cases} \delta_1(\cdot) & \text{if } c_i = 1 \qquad \textit{[complete view]} \\ TLL(\alpha, \beta) & \text{if } c_i = 0 \qquad \textit{[truncated view]} \end{cases}$

where $f_{TLL}(t; \alpha, \beta) = f_{LL}(t; \alpha, \beta)/Z$ is the PDF of right-truncated LL distribution on $t_i \in (0, 1)$, $Z = F_{LL}(t = 1; \alpha, \beta) - F_{LL}(t = 0; \alpha, \beta)$ for normalization, and $\delta_1(\cdot)$ denotes a point mass at 1.0.

The main idea behind UM-DP is that it considers separately the cases where (a) viewers make a preemptive choice to consume the complete media content (due to friendship, self-selection, etc.), and (b) viewers are less invested and drop off when they lose interest. Intuitively, this reflects a dichotomous choice in media consumption: sometimes, we “exploit” the media which we highly suspect to be interesting given factors like interest in the poster, subscriptions , fascination with a content thumbnail, etc., and other times we “explore” other content whom we give attention to in a fickle way. We note that UM-DP bears resemblance to *hurdle models*, which are often used to model over-inflation of 0s in ecologicaldata[238]; such models pose a “hurdle” via a Bernoulli probability, which when overcome allows a non-zero sample to be generated from an auxiliary process. Our UM-DP places such a hurdle of probability θ on $P(t = 1.0)$ to model complete views, and with probability $1 - \theta$ we sample from the auxiliary TLL distribution such that $t < 1.0$ for truncated views.

Inference of UM-DP. We infer parameters for UM-DP by maximizing the log-likelihood. The overall log-likelihood is given by

$$\ell(\theta, \alpha, \beta) = \sum \theta \log P(t_i = 1.0) + (1 - \theta) \log f_{TLL}(t_i; \alpha, \beta)$$

We can infer θ by maximum likelihood by taking the proportion of empirically observed complete views, i.e. $\hat{\theta} = \sum \mathbf{1}(t_i = 1.0)/n$ over n total views. After filtering the complete views, we can estimate the α, β parameters for TLL on the truncated views by maximizing likelihood heuristically as in the LM-DP case.

Validation of UM-DP. Again, we validate the model both qualitatively and quantitatively. Figure 9.6 illustrates parity between empirical data and superimposed model fits across unlooped media samples of varying durations, viewer counts and dwell time behaviors. We observe that our proposed UM-DP is able to well-approximate both the completed views (far right), and maintains good performance in modeling the peak/decay corresponding to viewer drop-off despite differences in distribution shapes.

Quantitatively, we again used KS tests and NLL to compare performance of UM-DP with alternatives: CU-LN (Log-normal), CU-IG (Inverse Gaussian), CU-WB (Weibull) and CU-G (Gamma). Technically, we used the truncated variants of these models in the same context

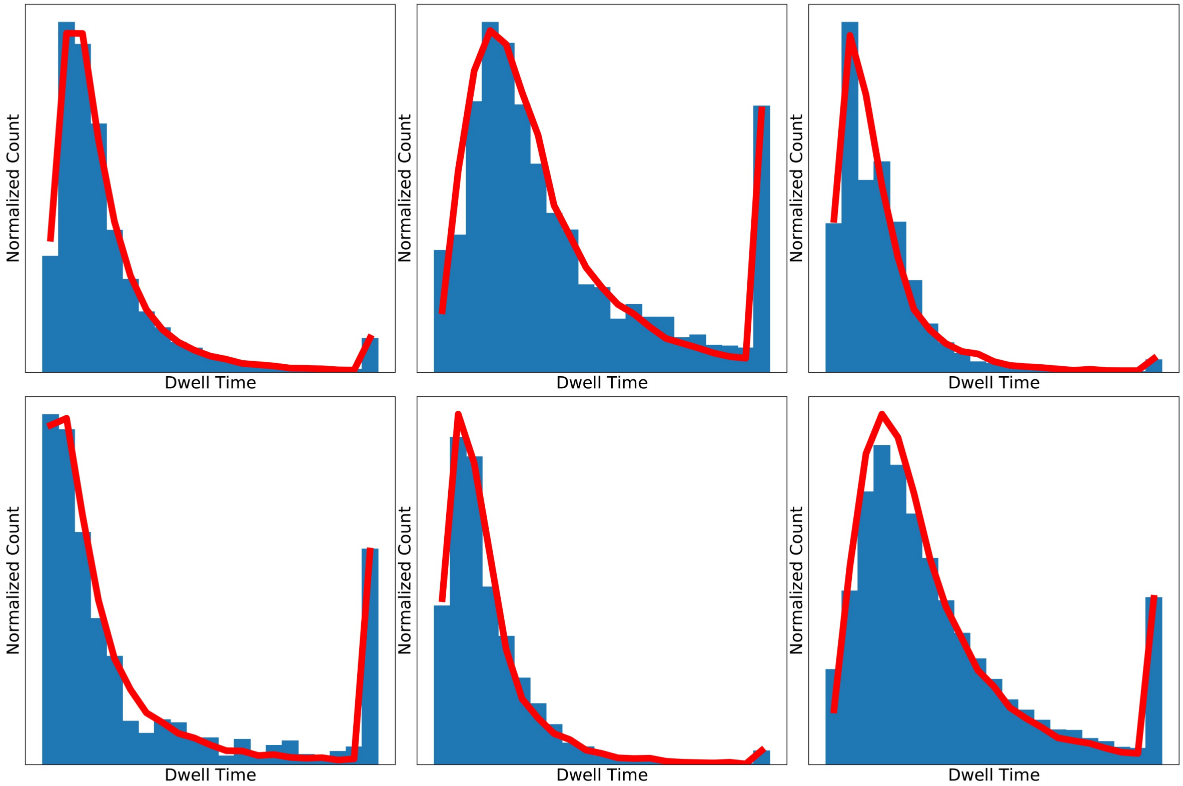
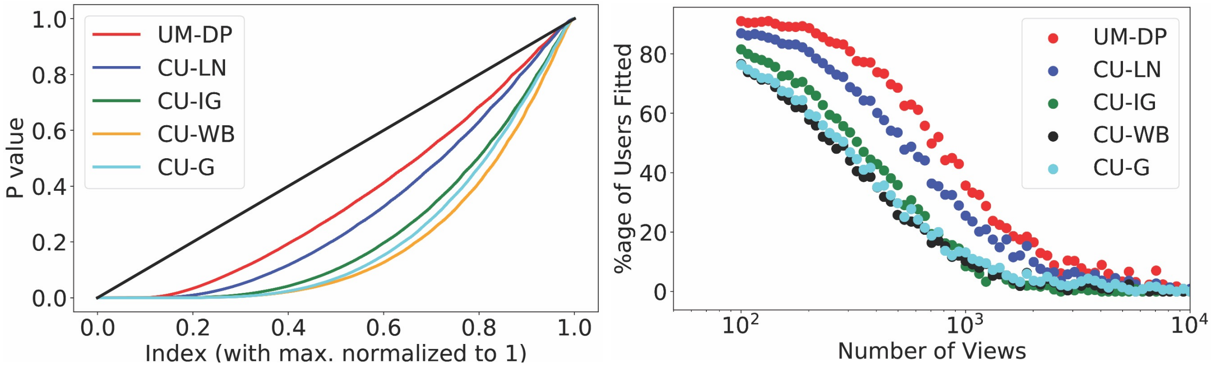


Figure 9.6: Our proposed UM-DP (red) visually matches empirical dwell time probabilities (blue) across unlooped media samples with varying viewing patterns.

proposed in our UM-DP formulation. Figure 9.7 shows the sorted p-values across samples in (a) and percentage of samples correctly fit against view count in (b); again, we observe that the proposed UM-DP fits the majority of samples well (around 90% with ≈ 100 views) outperforms



(a) Sorted p-values

(b) % of successful fits

Figure 9.7: UM-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the 45° line, better the fit. (b) %age of samples where model fits were successful($p < .05$).

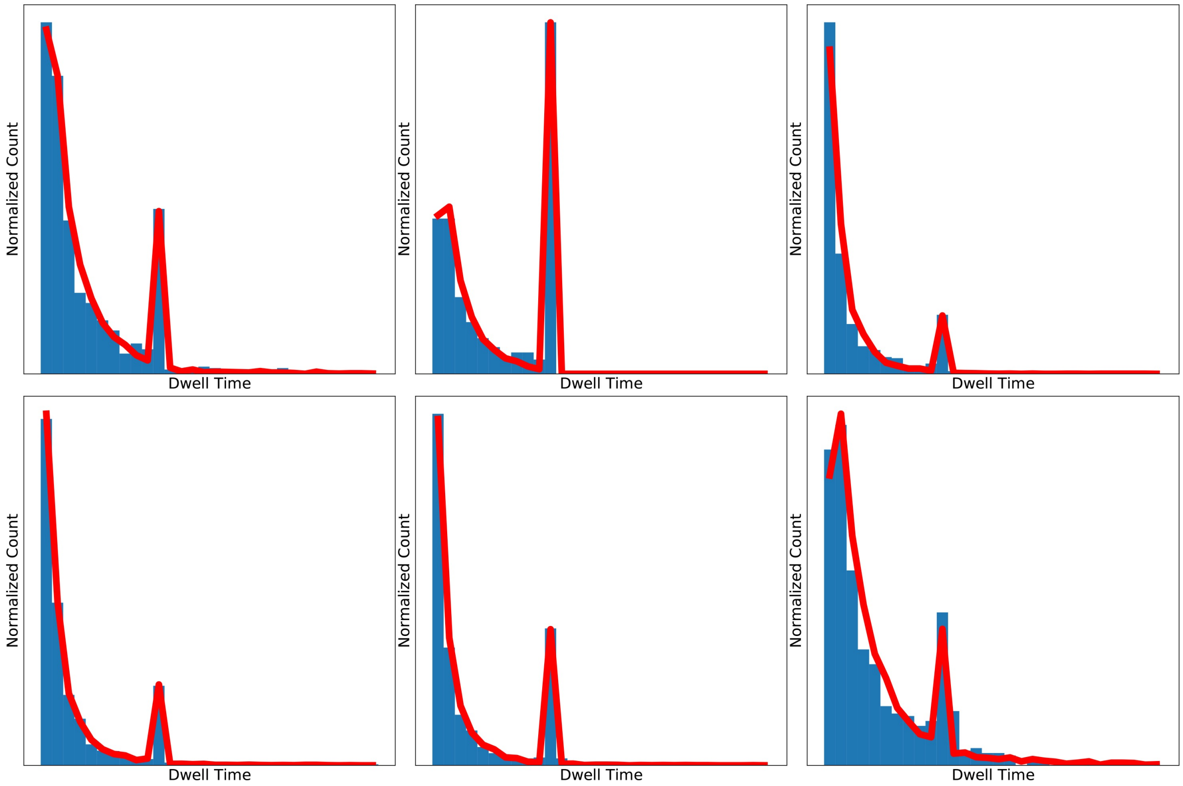


Figure 9.8: Our proposed V-DP (red) visually matches empirical dwell times (blue) across several looped media samples with varying viewing patterns.

the other models, with the CU-LN model the next closest, and CU-IG/CU-WB demonstrating significantly worse performance. Table 9.2(UM-DP) further shows that UM-DP outperforms the alternatives over aggregated percentage of samples better fit by both KS tests and log-likelihood comparisons across fitted models.

9.4.2 Viewers

Modeling viewers has a distinct set of challenges. Most notably, we must model viewers across time that they spend on looped and unlooped media both. Given the differences in support over dwell time ratios over the two, this is non-trivial. Moreover, we must account for differences in inherent propensities of viewers to watch unlooped and looped media. The alternatives to accounting for these complexities in a single joint model are undesirable, as they would result in having individualized models for each user across multiple content types and exposure durations, greatly increasing model complexity and requiring many more samples for inference.

To overcome these challenges, we propose V-DP (**V**iewer **D**well **P**rocess), which aims to unify the modeling of these heterogeneous phenomena. At the core of V-DP is the Log-normal distribution, which we observed closely matched the tails of many viewers' dwell time ratios. The Log-normal distribution is defined as

Definition 4 (Log-normal Distribution (LN))**.** *Let T be a non-negative continuous random vari-*

able, such that $T \sim LN(\mu, \sigma)$. The PDF and CDF of T are given by:

$$f_{LN}(t;\mu,\sigma) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(\log t-\mu)^2}{2\sigma^2}} \qquad F_{LN}(t;\mu,\sigma) = \Phi\left(\frac{\log t - \mu}{\sigma}\right)$$

where $t \in (0, \infty)$, $\mu \in (-\infty, \infty)$ and $\sigma > 0$ are the mean and standard deviation of $\log T$, and Φ indicates the standard normal CDF.

Like LL, the LN distribution is also commonly used in survival analysis [57]. Both distributions have very similar shapes; however, LL typically has heavier tails. Intuitively, this disparity in distributions between content-centric and viewer-centric modeling makes sense as viewers have more associated "outgoing" views than contents have "incoming" ones, and proportionally more of those views tend to be short. This would explain why viewer dwell time ratios exhibit more probability in the head of the distribution with lighter tails, making LN a more suitable option for the viewer modeling task than LL. Given this, we propose the V-DP as follows.

Definition 5 (V-DP)**.** *Sample each dwell time ratio t_i as*

1. $l_i \sim Bernoulli(\psi)$
2. $c_i \sim Bernoulli(\theta)$
3. $t_i \sim \begin{cases} LN(t;\mu_L,\sigma_L) & \textit{if } l_i = 0 & \textit{[LM view]} \\ \delta_1(\cdot) & \textit{if } l_i = 1, c_i = 1 & \textit{[UM comp. view]} \\ TLN(\mu_U,\sigma_U) & \textit{if } l_i = 1, c_i = 0 & \textit{[UM trunc. view]} \end{cases}$

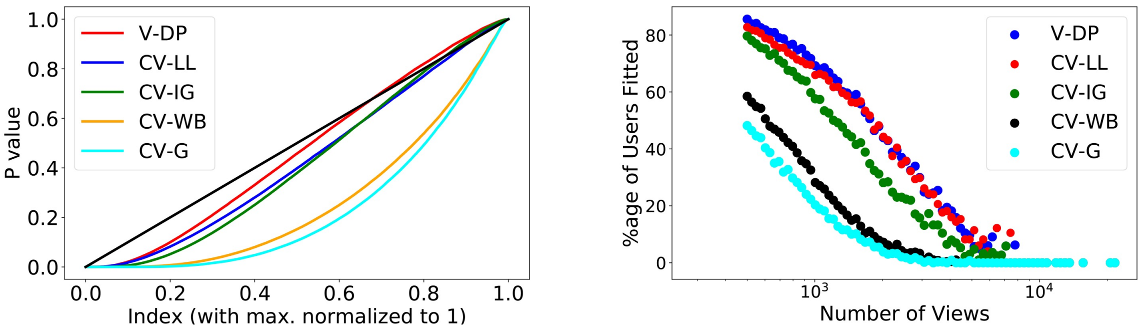
where $f_{TLN}(t;\mu,\sigma) = f_{LN}(t;\mu,\sigma)/Z$ is the PDF of right-truncated LN distribution on $t_i \in (0,1)$, $Z = F_{LL}(t=1;\alpha,\beta) - F_{LL}(t=0;\alpha,\beta)$ for normalization, and $\delta_1(\cdot)$ denotes a point mass at 1.0.

Our proposed V-DP is a mixture of viewing processes between both looped and unlooped content. The unlooped content has a max dwell time ratio of 1.0, and thus we sample views to this content in a manner similar to UM-DP, with the exception of using TLN distribution. Looped content has views with unbounded dwell time ratios, and thus we sample these views in a manner similar to LM-DP, but using LN distribution. The mixture proportions are determined by a parameter trading off propensity for looped versus unlooped media. Note that here, we model views to content with different exposure durations in the same, dwell time ratio model. Technically, though we describe the unlooped and looped views using a single LN variant each, we are actually observing the *convolution* of the underlying varying duration distributions.

Inference of V-DP. We aim to maximize the log-likelihood in inferring parameters for V-DP. The log-likelihood of is given by

$$\begin{aligned} \ell(\psi,\theta,\mu_U,\sigma_U,\mu_L,\sigma_L \mid t) = &\sum \psi[\theta \log P\,(t_i = 1.0) \\ &+ (1-\theta)\log f_{TLN}\,(t_i;\mu_U,\sigma_U)] \\ &+ (1-\psi)\log f_{LN}\,(t_i;\mu_L,\sigma_L) \end{aligned}$$

Consider n as the total number of views, and n_U and n_L as number of views on unlooped and looped content (such that $n_U + n_L = n$). Then, we have $\hat{\psi} = n_U/n$, and similarly if we consider the number of complete views on unlooped snaps as n_U^C, then $\hat{\theta} = n_U^C/n_U$. To infer parameters μ_L, σ_L for looped media views, we can use closed form estimators. To infer the LN parameters μ_U, σ_U for unlooped snaps, we maximize the TLN log-likelihood using Nelder-Mead.



(a) Sorted p-values

(b) % of successful fits

Figure 9.9: V-DP outperforms alternatives:(a) sorted p-values from KS tests; the closer a model curve to the $45°$ line, better the fit. (b) %age of samples where model fits were successful ($p < .05$).

Validation of V-DP. We validate V-DP both qualitatively and quantitatively. Figure 9.8 shows several example fits of V-DP on sample viewers; observe that our formulation allows a flexible fitting of various, complex distributional shapes which represent engagement with highly heterogeneous content using few parameters. Moreover, we compare V-DP with other candidate models, which as in UM-DP and LM-DP, differ from V-DP in the central parametric distribution used. The candidate models CV-LL (Log-Logistic), CV-IG (Inverse Gaussian), CV-WB (Weibull) and CV-G (Gamma), differing in replacement of LN distribution to respectively mentioned ones.

Quantitatively, we evaluate V-DP's goodness-of-fit by using KS tests and NLL. We plot the sorted p-values of V-DP and alternatives in Figure 9.9(a), demonstrating that V-DP's p-value curve is closest to the ideal and fits better than alternatives, with CV-LL coming in at a close second. Figure 9.9(b) shows the percentage of viewers that are successfully fit with V-DP; here too, we observe that V-DP fits for majority of the viewers (over 95% with ≈ 100 views) with CV-LL performing roughly on par at lower view counts, but trailing behind as view count increases. Again, decrease in fit performance at high view count is encountered by all models due to KS test sensitivity with large sample sizes. Table 9.2(V-DP) lists the aggregate percentage of cases where V-DP performs better than other candidate models in both KS p-values and NLL; NLL suggests significantly better fit performance using V-DP over the competitors.

9.5 Aggregate Dwell Time Modeling

Given parametric individual fits for each individual content or viewer sample, how can we identify patterns, normative behaviors and anomalies in dwell times of many content or viewer samples, respectively? How common is it to watch over 80% of an image or video? How common is it for a viewer's dwell times to be narrowly distributed around 5% and so on? To answer the above questions, we need to model the parameters in *aggregate*, across many samples. However, modeling the joint distribution of multivariate data is in general not trivial, posing challenges in dependency estimation, inference and curse of dimensionality. In this work, we propose to flexibly model joint distributions of parameters across many content and viewer samples respectively, using a powerful statistical tool known as a *copula* [217]. Copulas allow for scalable, parametric, approximate inference of multivariate distributions. This second level of parametricity in our modeling is advantageous, as it helps us better interpret inter-parameter dependency estimation, enables quick normality scoring and likelihood estimation, and moreover is generative, letting us actually simulate high-quality, realistic dwell time data.

9.5.1 Copula Modeling

Bivariate Modeling

Copulas are statistical tools, that explicitly model the dependency structure between given univariate marginals to estimate bivariate joint distributions. Copulas have been extensively used in finance [32], healthcare [211] and hydrology research [69]. We can define a bivariate copula as follows:

Definition 6 (Bivariate Copula)**.** *A bivariate copula C is a dependency function, defined as $C : [0,1]^2 \rightarrow [0,1]$. Given two random variables U and V and their marginal CDFs F_U and F_V, a copula $C(F_U(u), F_V(v))$ models the joint CDF, admitting a joint PDF of*

$$f_{U,V}(u,v) = f_U(u) \cdot f_V(v) \cdot c(F_U(u), F_V(v))$$

where c and f denote copula and marginal densities.

Technically, copulas are defined on uniform marginal CDFs. We can transform any random variable Y to uniformity by using probability integral transform (PIT) or vice-versa (inverse transform sampling). Various parametric forms of copula exist and can be used to capture different dependencies (positive, negative, independent) between different types of random variables. While bivariate copulas have demonstrated great empirical success in capturing dependencies via a variety of parametric forms, the number of generalized multivariate parametric copulas (for > 2 variables) are highly limited and inflexible in preserving pairwise dependencies, resulting in poor estimation. Given that some of our proposed models are multivariate, we seek a better option: to model multivariate dependencies parametrically while also allowing for flexible pairwise dependency modeling in high dimension, we propose the use of Vine copulas.

Multivariate Modeling

Vine copulas leverage the flexibility of parametric bivariate copulas to preserve bivariate statistical dependencies in higher-dimensional joint distributions. The dependency structure is modeled by the composition of (a) a set of bivariate copula families, (b) the associated copula dependency parameters, and (c) a nested tree structure to model the decomposition of joint distribution into the bivariate copula and marginal densities, as follows [53]

Definition 7 (Vine copula)**.** *A vine copula on n random variables $X_1 \dots X_n$ has a joint PDF defined by*

$$f_{X_1 \dots X_n}(x_1 \dots x_n) = \prod \prod c_{i,i+j|i+1,\dots,i+j-1} \cdot \prod f_k(x_k)$$

where c and f denote associated copula and marginal densities.

Different tree structures have been proposed to model these dependencies ; in this work, we use canonical vines (C-vines). C-vines decompose marginals and bivariate copula densities such that every tree has a one-to-many structure:

Definition 8 (C-vine)**.** *A set of linked trees $\mathcal{V} = (T_1, T_2, T_{n-1})$ is a C-vine on n elements if*

1. *T_1 is a tree with nodes $N_1 = 1, \ldots, n$ and a set of edges E_1 between a selected node $a \in T_1$ and all other nodes $b \in T_1$.*
2. *For $i = 2, \ldots, n-1$, T_i is a tree with E_{i-1} nodes and edge set E_i such that a single node in T_i is connected to all other nodes in T_i, and no other edges exist.*

Inference. To select the appropriate C-vine structure, we use the procedure as mentioned in [53]; specifically the node with maximum absolute Kendall's τ-correlation to other nodes is selected as central node for each level tree. Given the structure, we maximize log-likelihood to infer bivariate copulas and the associated parameters.

9.5.2 Multimedia Content

Below, we discuss how we conducted modeled aggregate modeling for looped and unlooped media.

Looped Content

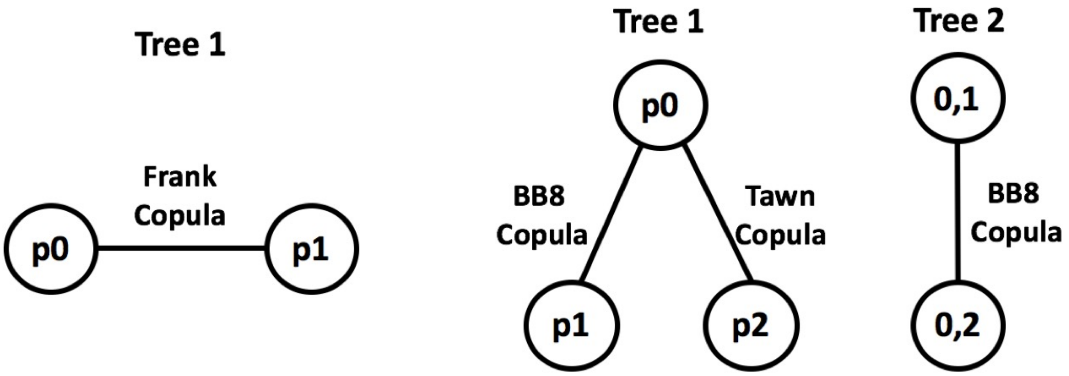
Since our LM-DP produces only 2 parameters for each content sample, a bivariate copula suffices to model the two-parameter dependency. To do this, we used LM-DP to to fit parameters for all looped media samples, and subsequently applied the PIT using the empirical CDFs for both α and β describing the dwell ratio scale and shape. We then selected the bivariate copula (shown in Figure 9.10(a)) which best maximizes the log-likelihood across a variety of parametric forms discussed in [214], and inferred parameters using 30% of the samples; we call this model LM-AM.

Unlooped Content

We obtained 3 parameters from UM-DP, θ, α, β which describe view completion rate and truncated view dwell time ratio scale and shape. Given the multivariate setting, we inferred parameters for a Vine copula (shown in Figure 9.10(b)), training on 30% of the samples as in the looped case. We call this model UM-AM.

9.5.3 Viewers

In modeling individual viewer dwell ratios, our V-DP produced 6 parameters for each viewer: $\psi, \theta, \mu_L, \sigma_L, \mu_U, \sigma_U$, denoting propensity to view unlooped media, propensity to complete unlooped views, and mean and standard deviation of the log dwell ratios for looped and truncated unlooped views, respectively. Using a sample of $100K$ instances , we estimated and fitted a C-vine. We call this model V-AM.



(a) LM-AM (Looped Media)

(b) UM-AM (Unlooped Media)

Figure 9.10: Bivariate and C-vine copula structures can model joint densities parametrically. (a) and (b) show our LM-AM and UM-AM dependency structures, respectively.

Table 9.3: Pearson correlation coefficients between parameters in original and simulated data.

Correlations	$(\mathbf{p0}, \mathbf{p1})$	$(\mathbf{p1}, \mathbf{p2})$	$(\mathbf{p0}, \mathbf{p2})$
Original	-0.32	0.08	0.43
Simulated	-0.31	0.10	0.41

9.5.4 Validation

To evaluate the performance of C-vine modeling in our usecase, we consider the following aspects:

- **Q1. Dependency preservation**: How well does C-vine approximate the original data dependencies?
- **Q2. Training size:** How is C-vine modeling performance influenced by training size?
- **Q3. Temporal consistency**: How robust is C-vine modeling for similar data from two different time-frames?

Given space constraints, we show experimental results only on UM-AM, noting that those for LM-AM and V-AM are similar.

Dependency preservation

Here, we determine if dependency structure in original data is well approximated by the C-vine model. To this end, we compare generated random samples from the simulated data on $[0, 1]^n$ ($n = 3$ for UM-AM, used here) to the PIT-representation of training samples. We report the pairwise Pearson correlations in Table 9.3, and show heatmaps of the pairwise dependencies in Figure 9.11. We observe that correlations between all parameter pairs and density estimates are closely approximated.

Training size

We also study the effects of training size on C-vine modeling performance. We experimented by training the C-vine model using random samples of varying sizes from the entire set, and sampling instances from the fitted models. To comparing samples from multivariate distributions, we use kernel-based Maximum Mean Discrepancy (MMD) test as proposed in [94] (the KS test is only suitable for univariate samples), to test the null hypothesis H_0: simulated samples and original data samples are from same distribution. We present the MMD test statistic for data simulated using models with various training sizes in Table 9.4; results show that we are not able to reject H_0 in any case. Even when using only 10% training data, we observe that the C-vine model closely approximates the original data distribution. Notice that the MMD statistic decreases as training size increases, showing closer approximation towards the original data.

Temporal consistency

We next aim to validate that aggregate models produced from C-vine are temporally consistent, in that they closely match across data taken from different time periods (we expect the underlying behavior does not shift significantly). We fit another C-vine model using dwell time engagement data from a different month than the data discussed here. We then compared the simulated data from both C-vine models and evaluated similarity between the two. To this end, we perform an MMD test between the two samples, obtaining a test statistic of 0.032, which does not let us reject H_0, and thus we can say they are drawn from same distribution. We observe this visually in Figure 9.12, where samples generated from both C-vines produce similar dependency structures for each pair.

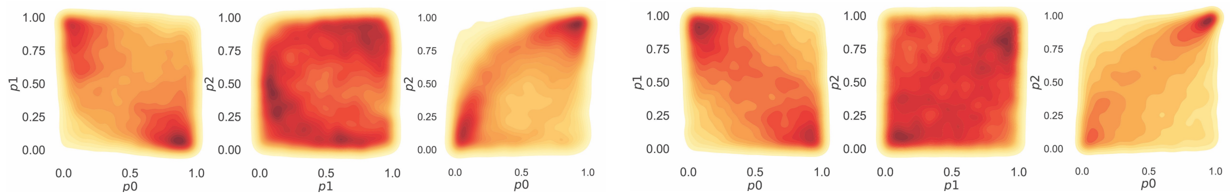


Figure 9.11: **Aggregate C-vine models closely approximate real data**. Pairwise dependency heatmaps between original data (top) and simulated data (bottom) are visually close.

Table 9.4: MMD test statistics between original data and model-simulated data (lower is better).

Training Size	10%	20%	30%	40%	50%
MMD-Statistic	0.037	0.034	0.0334	0.333	0.30
Reject H_0	No	No	No	No	No

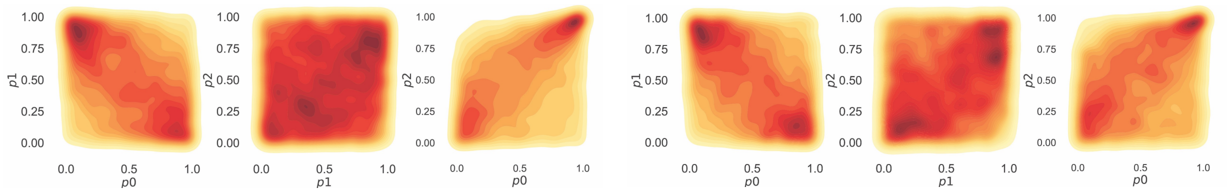


Figure 9.12: C-vine models are robust and consistent over time. Pairwise dependency heatmaps between simulated data from aggregate models trained on two different months (top and bottom) are visually close.

9.6 Anomaly Detection

In the previous section, we introduced parametric copula-based models for aggregate content and viewer modeling, demonstrating success in modeling the vast majority of samples while preserving complex interactions between parameters. A natural line of evaluation is determining effectiveness of such models in detecting anomalous engagement samples(i.e. samples which have extremely low-likelihood according to the aggregate models); thus, we pose the following questions.

- **Q1. Robustness to contamination**: Can our aggregate models robustly detect anomalies under contamination?
- **Q2. Qualitative efficacy**: Do our aggregate models detect real engagement anomalies on real data?

9.6.1 Robustness to contamination

We first study the performance of our aggregate model in detecting anomalies present in the training data, known as contamination. This scenario is possible in unsupervised models, like ours, as anomalous samples are not labeled and are also involved in individual and aggregate modeling steps. Ideally, our models should be able to detect anomalies in training data with high precision. To evaluate performance in such settings, we analyze our model's ability to successfully detect simulated attacks, by means of injecting artificial, anomalous samples in the original data. We present results for only Um-Am given space constraints, but results for Lm-Am and V-Am were similar.

We consider 4 different contamination models (shown below) in which anomalies are generated (a) according to different attack models, and (b) constituting varying contamination ratios. **Model 1 (Complete Views)**: Anomalies have all fully complete views: dwell time ratios of 1.0, **Model 2 (Long Views)**: Anomalies have overly long views: dwell time ratios sampled uniformly between 0.8-1.0, **Model 3 (Short Views)**: Anomalies have overly short views: dwell time ratios Gamma-distributed such that most dwell time ratios < 0.2, and **Model 4 (Uniform Views)**: Anomalies have random-length views: dwell time ratios sampled uniformly on 0.0-1.0. Also, we consider varying contamination ratios of 1%, 2% and 5% anomalies in the training data. We evaluate detection capacity via AUROC, which is reliable in imbalanced class settings like ours. Results are shown in Table 9.5, and indicate extremely high detection performance. We observe an AUC of 0.9+ across most scenarios, noting that higher contamination results intuitively result

Table 9.5: Anomaly detection performance (AUROC) under various anomaly contamination %ages (higher is better).

Attack Model	1%	2%	5%
Model 1 (Full Views)	0.99	0.98	0.96
Model 2 (Long Views)	1.0	0.99	0.99
Model 3 (Short Views)	1.0	0.99	0.98
Model 4 (Uniform Views)	0.94	0.92	0.84

in lower AUC due to increased model corruption.

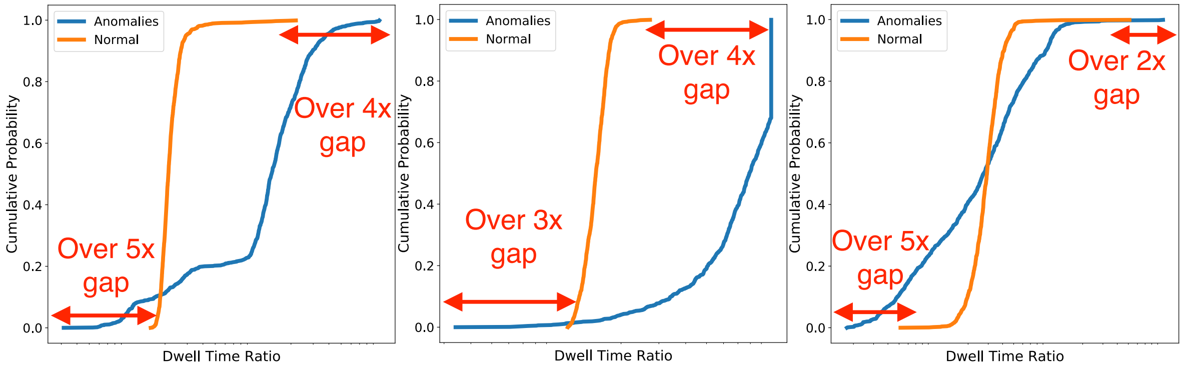
9.6.2 Effectiveness on real data

Next, we aim to evaluate whether our models can actually detect anomalous dwell time engagement in real data. To this end, we selected the 1000 most normal and anomalous samples according to log-likelihood, for each looped/unlooped content sample and viewer under LM-AM, UM-AM and V-AM respectively, and compared the empirical CDFs of mean dwell times across these entities. Intuitively, if our aggregate models were not detecting anomalous engagement, the empirical CDFs would closely match. However, as Figure 9.13 shows, the curves are significantly different for normal and anomalous samples identified by each model; note that the x-axis is in log-scale, making the observed differences more significant. We observe clear differences throughout the range of the CDF, and moreover discover the biggest differences near the extremities, suggesting our model does detects engagement anomalies. At the lower extremity of dwell time ratios, we observe that the lowest anomalous samples dwell times were $3 - 5\times$ smaller than those of their normal counterparts. Likewise, at the upper extremity, the highest anomalous sample dwell times were $2 - 4\times$ larger.

Manual inspection of several observed anomalous dwell time behaviors indicated significant abnormalities: (1) One anomalous viewer had over 5000 views/day, with mean dwell ratio < 0.03, and was adding more than 200 friends/day from an already staggering 3900, (2) several anomalous looped media samples with over 500 views had mean dwell ratio of $10 - 15\times$ the duration, and (3) several unlooped media samples with 100-300 views had mean dwell ratios of over 0.9; one sample with over 1000 views had a ratio of just 0.03. Figure 10(d) shows several examples of unlooped content anomalies discovered by UM-AM (others excluded for brevity). These anomalies could correspond to fake engagement, or possibly offensive or polarizing media. Overall, results demonstrate that our approach does empirically detect real-world anomalies across aggregate models, and could be additionally correlated with other features to discern abusive behaviors of various types.

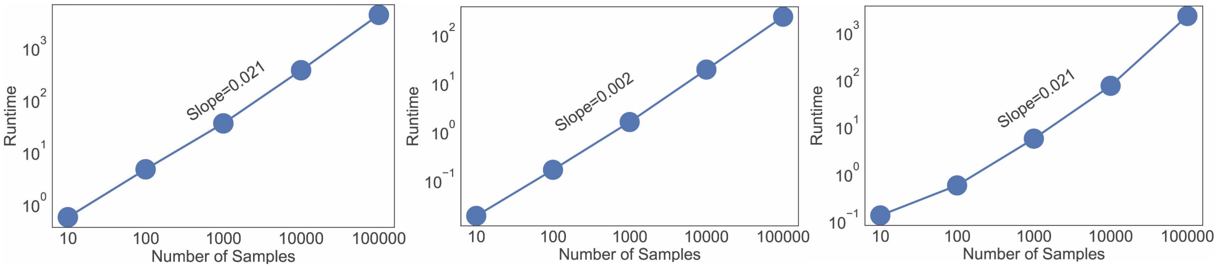
9.7 Scalability

We briefly discuss scalability in terms of both individual dwell process fitting and aggregate copula modeling. The major runtime cost in the former case is log-likelihood maximization for fitting parameters of the relevant dwell process.Figure 9.14(a) shows that this procedure exhibits



(a) Looped content (b) Unlooped content (c) Viewers

Figure 9.13: Our aggregate models detect real dwell time anomalies. The subplots show huge disparities in the mean dwell time ratio distributions between anomalous and normal (a) unlooped media, (b) looped media and (c) viewer samples.



(a) Individual fitting (b) PIT pre-processing (c) Copula inference

Figure 9.14: Our model inference is scalable: (a-c) show that individual fitting, copula preprocessing via integral transform, and copula inference are all near-linear in sample size.

empirically linear runtime in the number of training samples. The runtime costs in the latter case are incurred in conducting the PIT on original data samples for copula pre-processing, and selecting ideal copula structure and parameters. Figures 9.14(b) and 9.14(c) show that these steps admit linear and near-linear runtime respectively. Results are shown on UM-DP/UM-AM.

9.8 Conclusion

In this work, we provide the first comprehensive analysis of modeling dwell time engagement on visual multimedia content. Studying such content is valuable, as its consumption constitutes a significant portion of daily online activity, and has valuable applications in behavior modeling and anomaly detection. We first discuss challenges and considerations in the modeling task, including content heterogeneity and behavioral diversity. Our first contribution constitutes the LM-DP, UM-DP and V-DP generative dwell time processes and inference procedures, which enable *individual modeling* of content-centric and viewer-centric dwell time engagement. We show that these models match empirical data visually and quantitatively according to KS tests and outperform alternatives in both log-likelihood and KS tests. Our next contribution posits the analog LM-AM, UM-AM and V-DP, which enable *aggregate modeling* of joint distributions across individual fits using parametric bi/multivariate copulas. We demonstrate the flexibility of such models in capturing high dimensional dependencies with limited training data, show that they closely match original data both visually and quantitatively according to MMD tests, and are temporally consistent. Our last contribution includes ramifications of our proposed models for *anomaly detection*, in both robustness to contamination (0.9+ AUROC in most experiments) and qualitative evidence in terms of anomalies detected on real engagement data.

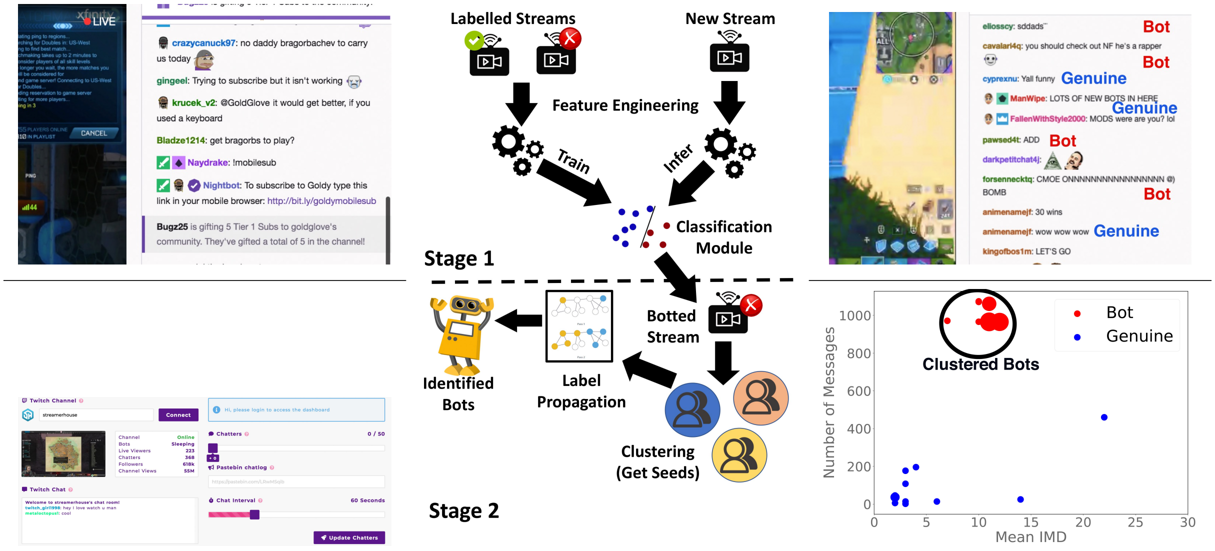
Shreya Jain, Dipankar Niranjan, Hemank Lamba, Neil Shah, Ponnurangam Kumaraguru. ”Characterizing and Detecting Livestreaming Chatbots”. 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2019.

CHAPTER 10

DETECTING CHATBOTS ON LIVESTREAMING APPLICATIONS

Livestreaming platforms enable content producers, or streamers, to broadcast creative content to a potentially large viewer base. Chatrooms form an integral part of such platforms, enabling viewers to interact both with the streamer, and amongst themselves. Streams with high engagement (many viewers and active chatters) are typically considered engaging, and often promoted to end users by means of recommendation algorithms, and exposed to better monetization opportunities via revenue share from platform advertising, viewer donations, and third-party sponsorships. Given such incentives, some streamers make use of fraudulent means to increase perceived engagement by simulating chatter via fake “chatbots” which can be purchased from shady online marketplaces. This inauthentic engagement can negatively influence recommendation, hurt streamer and viewer trust in the platform, and harm monetization for honest streamers. In this paper, we tackle the novel problem of automating detection of chatbots on livestreaming platforms. To this end, we first formalize the livestreaming chatbot detection problem and characterize differences between botted and genuine chatter behavior observed from a real-world livestreaming chatter dataset collected from Twitch.tv. We then propose SHERLOCK, which posits a two-stage approach of detecting chatbotted streams, and subsequently detecting the constituent chatbots. Finally, we demonstrate effectiveness on both real and synthetic data: to this end, we propose a novel strategy for collecting labeled, synthetic chatter dataset (typically unavailable) from such platforms, enabling evaluation of proposed detection approaches against chatbot behaviors with varying signatures. Our approach achieves .97 precision/recall on the real-world dataset, and .80+ F1 scores across most simulated attack settings.

In recent years, livestreaming platforms such as Twitch, YouTube Live, Facebook Live, and Ustream have grown to become dominant players in the content broadcasting space, commanding *millions* of broadcasters and *tens of millions* of daily active users [230]. These platforms provide avenues for broadcasters, or *streamers*, to share creative content of various forms (e-


crazycanuck97: no daddy bragorbachev to carry us today

gingeel: Trying to subscribe but it isn't working

krucek_v2: @GoldGlove it would get better, if you used a keyboard

Bladze1214: get bragorbs to play?

Naydrake: !mobilesub

Nightbot: To subscribe to Goldy type this link in your mobile browser: http://bit.ly/goldymobilesub

Bugz25 is gifting 5 Tier 1 Subs to goldglove's community. They've gifted a total of 5 in the channel!

Twitch Channel	
Channel	Online
Bots	Sleeping
Live Viewers	223
Chatters	368
Followers	618k
Channel Views	55M

Twitch Chat

Welcome to streamerhouse's chat room!
twitch_girl1998: hey I love watch u man
metaloctopus1: cool

Hi, please login to access the dashboard

Chatters 0 / 50

Pastebin chatlog: https://pastebin.com/LRwMSqib

Chat Interval 60 Seconds

elliosscy: sddads" — Bot

cavalari4q: you should check out NF he's a rapper — Bot

cyprexnu: Yall funny — Genuine

ManWipe: LOTS OF NEW BOTS IN HERE — Genuine

FallenWithStyle2000: MODS were are you? lol

pawsed4t: ADD — Bot

darkpetitchat4j:

forsennecktq: CMOE ONNNNNNNNNNNNNNNNN @) BOMB — Bot

animenamejf: 30 wins

animenamejf: wow wow wow — Genuine

kingofbos1m: LET'S GO

Figure 10.1: (**Left**) Livestreaming platforms offer chatrooms (top), which streamers can manipulate via chatbotting tools (bottom) that enable customization of chat interval, number of chatters and even message contents. (**Center**) We propose SHERLOCK, a two-stage chatbot detection approach based on stream (top) and user-level classification (bottom). (**Right**) We enable discovery of chatbotted streams (top – notice genuine users asking for moderators to handle the bots), and the constituent chatbots via discriminative features (bottom – large points indicate high user density).

sports gameplay, live events, art, etc.) to a large audience. Each broadcasting session, or *stream*, consists of two key components – the content being shared live to viewers, and a chatroom (Figure 10(left)), where viewers can chat and interact amongst themselves, and with the streamer. These chatrooms provide a completely different community experience to viewers in contrast to traditional media, providing an increased sense of participation and gratification [49].

Most livestreaming platforms recommend streams to would-be viewers based on prior and current engagement metrics, which is effectively a function of viewership and chatroom activity. Specifically, streams that garner high viewership and have active chatrooms are considered to be likely interesting and engaging to new viewers, and are thus recommended to draw new viewers, amplifying preferential attachment effects. Moreover, streamers who produce such content and draw such engagement are prime candidates for on-platform and off-platform monetization via advertising revenue share, donations from viewers, and sponsorships from third-parties (i.e. computer hardware companies for e-sports professionals). Such incentives lead some streamers to resort to fraudulent methods to increase their viewership [191] and increase chatroom activity [58]. Numerous online marketplaces like `streambot.com` and `youtube-livebot.com` offer streamers the ability to increase their chatroom activity over sustained period of time, via *chatbots* which simulate human-like chatter. Such fraudulent engagement has several adverse effects: (a) honest streamers may not be as highly recommended as fraudsters and lose out on potential engagement they may have otherwise garnered via preferential attachment, (b) viewers and streamers have reduced trust in the platform to recommend and prioritize good content, (c) the platform and third-party sponsors may lose money by partnering with fraudulent streamers who reach much lesser human eyes than their metrics suggest. Despite these concerns, prior work in mitigating chatbot abuse on livestreaming platforms is minimal – we seek to bridge the gap in this work.

There are numerous challenges in this problem setting: (a) *noisy data*: livestreaming chatter is full of messages with ill-formed sentences, containing spelling errors, "legitimate" spam messages (copypasta), and emotes, limiting efficacy of text-based features to identify fraudulent activity, (b) *user-controlled fraud*: most chatbotting services available on online marketplaces allow streamers to control the bots (Figure 10), giving them the ability to decide when and how much fake chatter should be introduced, and thereby complicating the attack space and hurting detection generalizability, and (c) *lack of ground-truth*: as livestreaming platforms operate at an extremely large-scale and do not reveal the chatbots they proactively ban from the service, obtaining reliable ground truth for building machine learning models is non-trivial.

In this work, we tackle these challenges and more. To our knowledge, we are the first to study the chatbot detection problem in the livestreaming setting. Specifically, our contributions are as follows:

1. **Problem formulation:** We formalize the chatbot detection problem in the context of chatrooms of livestreaming platforms.
2. **Dataset collection and characterization:** We obtain real livestreaming chatlog data, and compare the behaviors of chatbots and real users. We also discuss how to construct *labeled* synthetic chatter datasets from livestreaming platforms, for a variety of attack models.
3. **Proposed framework:** We propose SHERLOCK which tackles chatbot detection in a two-stage approach: detecting botted livestreams using a classification model (stage I), and detecting constituent chatbots using a seeding and label propagation approach (stage II). Overview of approach given in Figure 10(center).

We conduct several experiments to demonstrate that our proposed method is (a) *effective*: we show that our approach outperforms alternatives in detection performance on real chatlog datasets (.97 precision/recall) (Figure 10(bottom-right))(b) *robust to different attacks*: we show consistently good performance in detecting chatbots across many attack configurations ($\geq$ .80 F1 against most attack settings), and (c) *scalable*: our approach scales near-linearly on large datasets, especially due to our two-stage task formulation. We make the code for SHERLOCK available at `https://github.com/shreya-03/Sherlock`.

10.1 Related Work

We discuss prior work in (a) detecting chatbots, and (b) astroturfing in social media.

Detecting chatbots. Most prior work on chatbot detection consider chatbots as accounts that spread malicious or spammy URLs [84, 98, 196]. [84] proposed a classifier based on entropy-based features (message length, and inter-message delay) to detect chatbots on Yahoo chat systems. [196] used similar features to differentiate between bot and genuine users on various instant messaging platforms in IM (instant messaging) settings (i.e. human is chatting only with one user (bot/genuine)). Additionally, [98] proposes detecting chatbots based on the links they post, using cues from spam classification literature to detect malicious URLs. However, all of these methods are based on IM platforms, where chat messages are more directed towards other chatters, and are primarily concerned with delivering a payload of a malicious URL. Our work studies chatbots on livestreaming platforms, where bots are used with an alternative purpose of increasing perceived chatter, and hence vary in their design and motive. Though many works tackle bot detection on popular social media platforms, such as Twitter [253, 255], Facebook [22], and software marketplaces [12, 163], they are characteristically different from our work as they do not focus on detecting chatbots. Besides this, there is a lot of work in designing conversational agents [144], which is beyond the scope of this work as they aim at coming up with creating realistic chatbots not for malicious purposes.

Astroturfing in social media. Social media websites have become a common target for astroturfing, where users artificially inflate engagement to increase perceived popularity. Graph-based factorization approaches to group nodes based on similarity or dense connectivity implying suspicious, large clusters have shown considerable success in detecting fraudulent activities [22, 130, 254]. Random-walk based methods have also been used to detect abnormal cuts between suspicious and legitimate parts of a social graph [309]. Content-based methods use textual features [12] or local engagement features (i.e. based on egonets) [11] to detect spam and fraud. [70] also propose temporal methods focusing on finding anomalous patterns in multivariate time series. The closest work to ours is by [252], in which the author proposes an unsupervised method to detect livestreaming viewbots. Despite rich literature in this space, none of the prior works have focused on the problem setting of detecting chatbots on livestreaming platforms.

10.2 Problem Statement

Each stream on a livestreaming platform generally consists of a chatroom panel located adjacent to the live video player (Figure 10(Left)). Viewers must be signed in to participate in chat, and the messages typed by any of the signed-in viewers appears in realtime as the user sends each message. Each message is associated with the username of the author, as well as its timestamp. All chat messages are textual (i.e. text, emojis, URLs). Messages are typically short, and have a length cap to prevent single users from dominating the community chatroom with spam. Such chatrooms typically allow users to reply to one another (via an "@handle" mechanism), inducing a conversational aspect to the room. In this work, we leverage all available sources of information above: Specifically, we collect data pertaining to a set of livestreams $\mathcal{S}$. For each stream $s \in \mathcal{S}$, we collect the set of all messages $\mathcal{M}_s$. We refer to messages on stream s that were posted by user/chatter i as $\mathcal{M}_{s,i}$, and the timestamp of the jth message from user i on stream s as $t_{s,j,i}$. Given these information sources, we aim to detect chatbots.

We note that considering all users in all streams is a computationally heavy and expensive task. Moreover, it is difficult to claim in isolation whether any given message is from a chatbot or real user, and even if a single user is a chatbot or not. We take a step back to consider that instead of gauging whether each message or user is legitimate or not, we should first consider the aggregate behavior of the parent stream. This is because it is unlikely to observe a single chatbot in isolation, but far more likely to observe a number of chatbots orchestrating a coordinated

Table 10.1: Dataset Statistics

# of chatlogs	690
# of messages	$439,650$
# of streamers	168
# of chatters	$8,885$
Median stream duration	2.7 hours

activity inflation effort on a given stream. By focusing on a stream-level first, we can leverage aggregate behaviors from many messages from many users jointly to infer whether the stream appears to be botted or not. We formally define this task as follows:

Problem 1 (**Chatbotted Stream Identification**). ***Given*** *a set of streams $\mathcal{S}$, and corresponding set of chatters $\mathcal{C}_s$ for each $s \in \mathcal{S}$,* ***find*** *the set of chatbotted streams.*

Upon obtaining the set of suspected chatbotted streams $\mathcal{S}_{cb}$, we can next focus only on this subset to discern suspected chatbots from real chatters. We argue that while it is conceivable that chatbots may exist in isolation in other streams, it is unlikely, and at best ineffective from the streamer's point of view. Moreover, since $\mathcal{S}_{cb}$ is likely to be much smaller than $\mathcal{S}$, we can dramatically improve scalability by avoiding chatbot detection for determined "low-suspicion" streams, and only focusing on the high-suspicion ones. The task that we pose for these is as follows:

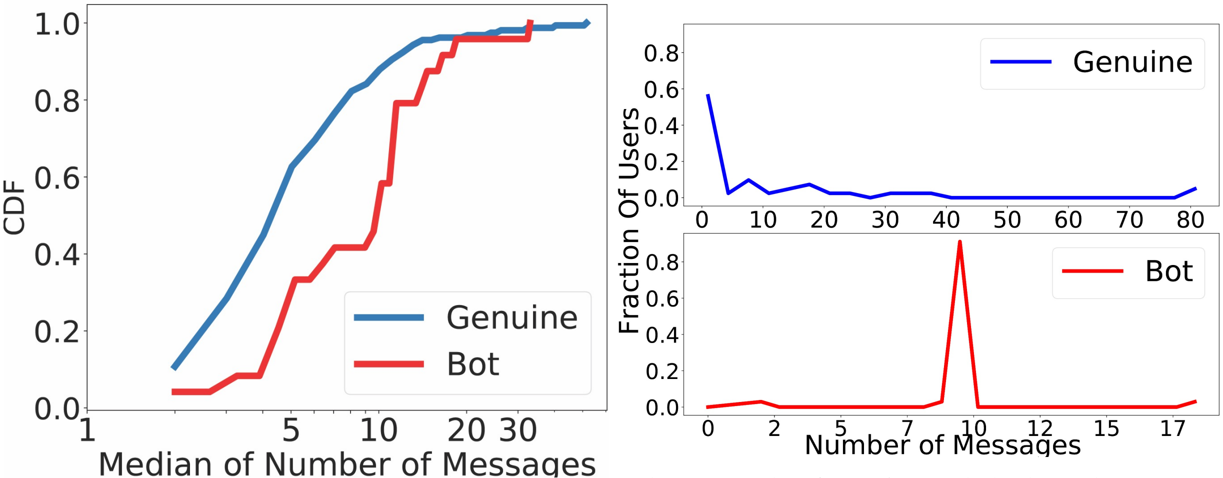
Problem 2 (**Chatbot Identification**). ***Given*** *a suspicious chatbotted stream $s \in \mathcal{S}_{cb}$, and corresponding set of individual chatters $\mathcal{I}$,* ***label*** *each chatter $i \in \mathcal{I}$ as being part of the (disjoint) set of real users $\mathcal{I}_r$ or chatbotted users $\mathcal{I}_{cb}$.*

10.3 Data Description

In this work, we study Twitch, a dominant livestreaming platform with over $2.2M$ streamers and $15M$ unique daily viewers reported in 2018[1]. Note that due to limitations on data collection and labelling cost, it is unfeasible to work with their platforms. However we assume that a similar method of providing chatbots is also used for other livestreaming platforms. We collected chatter of $439K$ messages over a period of three months from August to October 2018 from chatrooms of 690 randomly chosen Twitch streams. A brief description of the dataset collected is given in Table 10.1.

Annotation. We manually annotated 183 chatlogs out of the 690 collected. The annotators used cues such as relevance of text to the context, number of messages posted by accounts, metadata and other similar signals to identify if a particular livestream was chatbotted or not, as per knowledge from prior literature [252] and a survey of chatbotting services. The annotators found 24 botted and 159 seemingly genuine streams. While annotation was possible, it took each annotator roughly 104 hours to complete the task, clearly making annotation of the entire dataset infeasible. Thus, for our further analysis, we use the 183 streams, with $78,124$ messages from $6,167$ genuine users and $23,236$ messages from $2,739$ chatbots.

[1] `https://twitchadvertising.tv/audience/`



(a) Number of messages distribution ECDF.

(b) Example of genuine and chatbotted streams.

Figure 10.2: (a): ECDF for median distribution on number of messages for genuine and chatbotted streams. (b): Distribution of number of messages posted for randomly selected genuine and chatbotted streams.

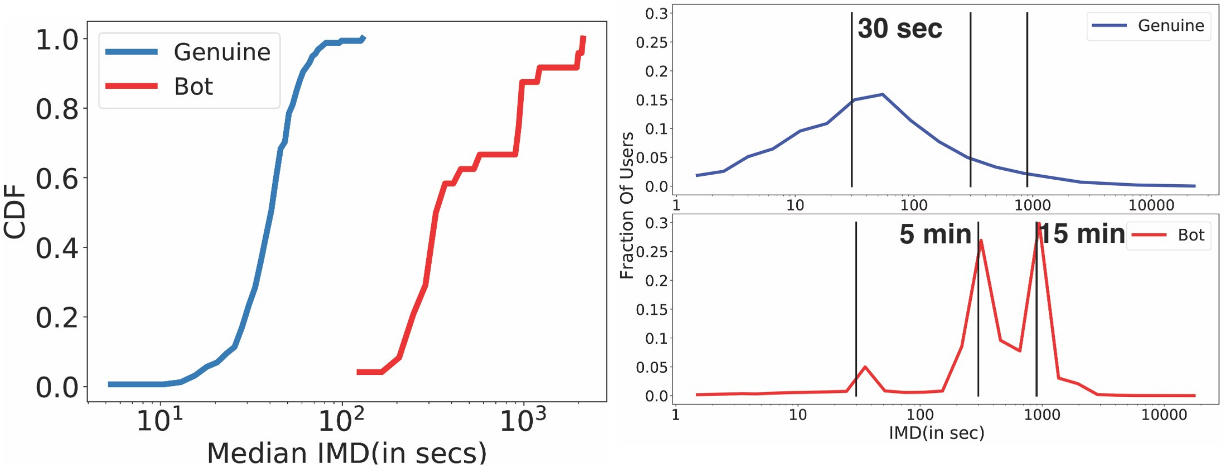
10.4 Initial Observations

Before proposing our approach, we conduct preliminary exploration of the dataset and try to identify key statistics that can help us differentiate the genuine and suspicious streams/bots. In this section, we describe the potential features we considered and point out the key insights we obtained about genuine and fraudulent behavior.

Message frequency. Since bots are created with the purpose of increasing chatroom activity, it is natural to assume that they will post more messages than the genuine users in a stream. However, it could be contrary as well, that is if the users are fooled by the bots into believing that bots are actually genuine accounts, it might happen that genuine users might keep up the end of conversation and end up creating similar or more messages than the bot accounts. For each stream, we compute the median of the number of messages posted distribution. We observe that the number of messages for chatbotted streams is higher than that of genuine streams. We show this by plotting the empirical cumulative frequency distribution (ECDF) in Figure 10.2(a). Additionally, we observe that the median statistic is able to differentiate between chatbotted streams and genuine streams with a Kolmogrov-Smirnov test p-value of 4.34×10^{-8}. Based on the above statistics, we make the following key observation:

Observation 3 (**MESSAGE FREQUENCY**). *Chatbots tend to post more messages than genuine users, with most chatbots posting messages with similar frequency.*

Inter-message delays (IMD). IMDs have been used previously in literature to identify bot behavior [70]. They have proved to be useful in identifying footprints of automation by scripting, which tends to be regular and deterministic. We define IMDs for an entire stream as the difference in time between each pair of consecutive messages from the same user, across all users for the duration of the stream.



(a) Median IMD distribution ECDF.

(b) IMD distribution for all genuine (top) and botted (bottom) streams

Figure 10.3: (a): ECDF for distribution of median on IMD for genuine and chatbotted streams. (b): Distribution of IMD.

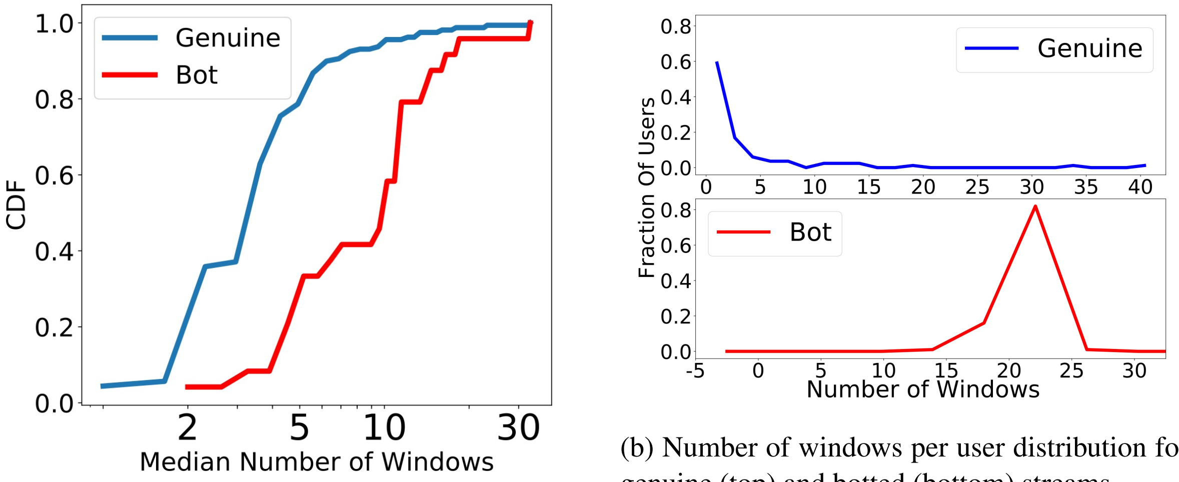
We plot the ECDF of median IMDs for each stream in Figure 10.3(a). We can observe that ECDF differs significantly for genuine and chatbotted streams (KS Test p-value: 1.93×10^{-19}). We also plot the PDF across all IMD for users in genuine streams and users in botted streams, and show this in Figure 10.3(b). Based on the above plots, we make the following observation:

Observation 4 (**INTER-MESSAGE DELAYS**). *Chatbotted streams have a higher IMD than genuine streams. Chatbots have a consistent IMD showing that they are automated.*

Message Spread. Since bots are designed to maintain engagement for extended periods (rather than specific times), we hypothesize that they post throughout the duration of most chatbotted streams. We investigate this empirically by counting the number of equal-duration time intervals in which a particular user posts during the duration of the stream. To compute this, we partition the stream into equal-duration intervals, and count the number of windows in which each user posts a message. Intuitively, users who post consistently will appear in more windows. Figure 10.4(a) shows the ECDF of the median of the number of windows per user distribution. We note that the distribution for chatbotted and genuine stream is significantly different, corroborated by a KS test with p-value of 7.34×10^{-7}. Figure 10.4(b) shows examples of these distributions for a chosen bot and genuine stream. We have the following observation:

Observation 5 (**MESSAGE SPREAD**). *Chatbots' message distribution is more spread out, and on average, they post consistently throughout the stream.*

Textual Cues. We additionally experimented with various features from text mining literature to determine if language used by chatbots is significantly different from that used by genuine users. We compared tf-idf scores, conversational dynamics, and similarities in term usage within chatbot and genuine user groups. Interestingly, we were not able to find any distinguishing patterns for chatbots. This is likely due to (a) message text being extremely noisy and short, (b) too much sparsity for conversation threads via "@handle" mechanism, and (c) most chatbot marketplaces enable customers to upload a text file of quotes used by chatbots, making the text



(a) Median number of windows distribution ECDF.

(b) Number of windows per user distribution for all genuine (top) and botted (bottom) streams

Figure 10.4: (a): ECDF for distribution of median on number of windows per user for genuine and chatbotted streams. (b): Distribution of number of windows per user.

customizable and seemingly relevant to legitimate chatter on the stream. An illustrative example of bot messages being short, noisy and indistinguishable from genuine messages is shown in Figure 10(top-right).

10.5 Proposed Framework: SHERLOCK

We next propose SHERLOCK, a two-stage framework which solves Problems 1 and 2 as discussed below.

10.5.1 Stage I: Detecting Chatbotted Streams

Given the set of streams $\mathcal{S}$, we first aim to detect the chatbotted streams $\mathcal{S}_{cb}$. Based on observations from Section 10.4, we aim to featurize streams in a space that can best differentiate chatbotted and genuine streams. We discuss our features below:

Number of messages. Observation 3 shows that chatbotted streams tend to have higher numbers of messages than genuine ones. Though many summary statistics can be extracted from the number of messages distribution, we found that weighted top-k modes (most frequent values) worked well empirically, as they represented the k largest "peaks" in the distribution. We were interested in capturing (possibly multiple) spikes in the distribution (for example, see Figure 10.2), which are generally associated with chatbotting activities. We used $k = 3$ to avoid introducing noise. Further, we weighed each of the k peaks with the associated fraction of users, allowing us to capture the intensity and overall contribution of the peak. Intuitively, peaks at large number of messages, and with high fraction of users are the most suspicious. This produces 3 features.

IMD quantiles. Observation 4 reflects that chatbotted streams tend to have higher IMDs than genuine ones. Moreover, many chatbots have spiky behavior which involves long lulls between chat messages. To capture the spikes and the overall higher IMD of chatbots, we used higher quantiles of the stream IMD distribution ($60\%^{ile}$, $70\%^{ile}$, $80\%^{ile}$, $90\%^{ile}$).

Number of windows. Observation 5 posits that since chatbots send messages atypically, and spread throughout the chat (rather than in quick conversations), they appear in higher numbers of windows than genuine users. Thus, a stream with many chatbots will likely have a number of window distribution with peaks associated with chatbot behaviors. Following the same rationale as before, we take the weighted top-k modes, again using $k = 3$ to avoid noise.

Concatenating these, we arrive at a 10-dimensional feature space. Next, we train a supervised model over this feature space and use the classifier to predict chatbotting propensity for any new, unseen stream. We add those with a sufficiently confident predictions to $\mathcal{S}_{cb}$.

10.5.2 Stage II: Detecting Constituent Chatbots

Upon obtaining a set of chatbotted streams $\mathcal{S}_{cb}$, our goal for each stream $s \in \mathcal{S}_{cb}$, is to label each user $i \in \mathcal{I}$ (relevant chatters) as belonging to real users $\mathcal{I}_r$ or chatbots $\mathcal{I}_{cb}$. We use a semi-supervised learning approach for this stage; such approaches have been demonstrably useful in tasks for which ground truth is limited. In the livestreaming case, collecting ground-truth for individual users as chatbots is highly challenging, time-consuming and unscalable. Thus, we employ a label propagation approach to identify chatbots.

Generating seeds. The success of our label propagation approach for classifying users naturally depends on the goodness of the seed labels. If a stream $s \in \mathcal{S}_{cb}$ has a sufficiently high prediction score, we conjecture that $\mathcal{I}_{cb}$ will be large compared to $\mathcal{I}_r$. With this key assumption, we consider certain regions of our feature space to identify *seed users* for whom we have "high confidence" *seed labels*. We use heuristics based on our earlier observations to obtain these seed labels. Specifically, our approach begins by bootstrapping seed sets using empirically observed highly discriminative features (i.e. high confidence seeds):

Number of messages. Observation 3 notes that chatbots tend to post more messages than genuine users. We denote number of messages sent by chatter i as $\mathbf{m}_i$.

Mean IMD. Observation 4 notes that chatbots tend to have longer IMDs than genuine users. We denote chatter i's mean IMD as $\mathbf{d}_i$.

Subscription status. Many livestreaming platforms offer paid subscription models, where users can pay to subscribe to a streamer. We assume that subscribers are genuine chatters, and can thus be exonerated. We use $\mathbf{r}_i$ to indicate chatter i's subscription status.

Next, we refine the seeds by exploiting synchronicity over less discriminative features to gain confidence in seed veracity; we use the following features:

Number of windows. The message spread of a chatter provides a strong signal if a particular chatter is a bot or not. We count the number of unique windows a chatter i posts a message in and denote it by $\mathbf{w}_i$.

IMD entropy. In addition to computing mean IMD, we also compute entropy of IMD. For each chatter i, entropy of it's inter message delay distribution is given by $\mathbf{h}_i = H(\text{IMD}_i)$.

This approach is summarized in Algorithm 2, which we describe next. We first consider

Algorithm 2 SEEDUSERS

```
Require: Number of messages vector, $\mathbf{m}$, mean IMD vector $\mathbf{d}$, number of windows vector $\mathbf{w}$, IMD entropy vector
    $\mathbf{h}$, subscriber indicator vector $\mathbf{r}$, synchrony threshold $n_{sim}$
Ensure: Refined seed sets $\mathcal{R}_{cb}$, $\mathcal{R}_r$
 1: Project all users into a subset feature space: $\{\mathbf{m}, \mathbf{d}\}$
 2: Remove outlier chatters in this subset feature space.
    ▷ Initialize candidate bot region
 3: $\mathcal{R}_{cb} \leftarrow \{i \in \mathcal{I} \mid \mathbf{m}_i > \mu(\mathbf{m}) \text{ and } \mathbf{d}_i > \mu(\mathbf{d})\}$
    ▷ Initialize candidate genuine user region.
 4: $\mathcal{R}_r \leftarrow \{i \in \mathcal{I} \mid \mathbf{m}_i < \mu(\mathbf{m}) \text{ and } \mathbf{d}_i < \mu(\mathbf{d})\}$
    ▷ Exonerate users with paid subscriptions.
 5: $\mathcal{S} \leftarrow \{i \in \mathcal{I} \mid \mathbb{1}(\mathbf{r}_i)$
 6: $\mathcal{R}_{cb} \leftarrow$ (largest cluster in $\mathcal{R}_{cb}$) $\setminus \mathcal{S}$
 7: $\mathcal{R}_r \leftarrow$ (largest cluster in $\mathcal{R}_r$) $\cup \mathcal{S}$
    ▷ Track # windows and IMD entropy in candidate bot region.
 8: Create bounding box $\mathbf{B}_{cb}$ around cluster $\mathcal{R}_{cb}$.
 9: $\mathcal{W} \leftarrow \{\}$                                   ▷ multiset with freq. $m_W(\cdot)$
10: $\mathcal{H} \leftarrow \{\}$                                   ▷ multiset with $m_H(\cdot)$
11: for chatter $i$ in $\mathbf{B}_{cb}$ do
12:     $\mathcal{W} \leftarrow \mathcal{W} \cup \{w_i\}$
13:     $\mathcal{H} \leftarrow \mathcal{H} \cup \{h_i\}$
    ▷ Augment chatbot seeds with too-synchronous users.
14: $\mathcal{W}_{sync} \leftarrow \{w \in \mathcal{W} \mid m_W(w) \geq n_{sim}\}$
15: $\mathcal{H}_{sync} \leftarrow \{h \in H \mid m_H(h) \geq n_{sim}\}$
16: for chatter $i \in \mathcal{I}$ do
17:     if $w_i \in \mathcal{W}_{sync}$ and $h_i \in \mathcal{H}_{sync}$ then
18:         $\mathcal{R}_{cb} \leftarrow \mathcal{R}_{cb} \cup \{i\}$
19:         $\mathcal{R}_r \leftarrow \mathcal{R}_r \setminus \{i\}$
20: Return $\mathcal{R}_{cb}, \mathcal{R}_r$
```

consider all users in $\mathcal{I}$ on $\mathbf{m}$ (number of messages) and $\mathbf{d}$ (mean IMD) (Line 1), as we empirically observed that these features are highly discriminative. In this $(\mathbf{d}, \mathbf{m})$ space, we first remove outliers (Line 2) in sparse regions due to low confidence about their status. Next, we initialize sets $\mathcal{R}_{cb}$ and $\mathcal{R}_r$ with users who have jointly high, and jointly low values on the features; these sets represent candidate bots, and candidate genuine chatters respectively (Lines 3-4). For users in each $\mathcal{R}_{cb}$ and $\mathcal{R}_r$, we next identify the largest cluster of candidate bots and genuine users (we use X-Means clustering [225] as it automates choice of cluster count using information theoretic measures), and add them to the seed set with respective labels (Lines 6-7). We further refine the seeds by exonerating users where $\mathbb{1}(\mathbf{r}_i)$.

Next, we refine $\mathcal{R}_{cb}$ and $\mathcal{R}_r$. To do so, we first construct a bounding box $\mathbf{B}_{cb}$ around $\mathcal{R}_{cb}$ (Line 8), which captures nearby users that may be missing in $\mathcal{R}_{cb}$, but may still be suspicious. We then consider the number of windows $\mathbf{w}$ and IMD entropy $\mathbf{h}$ feature values for these users, as we empirically observed that many chatbots tend to share similar values (motivated by Observations 4-5). We identify the feature values that occur over users in $\mathbf{B}_{cb}$ with greater than a given frequency n_{sim} as supposed "peaks" or bot signatures. Given these, we add chatters in $\mathcal{I}$ who have highly recurring feature values to $\mathcal{R}_{cb}$, and also remove them from $\mathcal{R}_r$ if applicable. In effect, our seeding process is a two-level clustering, where the first-level relies on exploiting knowledge of suspicious regions in the $(\mathbf{d}, \mathbf{m})$ space, and the second-level relies on augmenting this with non-region-specific synchronicity in the $(\mathbf{w}, \mathbf{h})$ space. We note that we considered seeding via a single clustering stage in experimentation, but achieved poor results due to noisiness induced by the less-discriminative features.

Propagating suspiciousness. Upon obtaining the seed sets $\mathcal{R}_{cb}$ and $\mathcal{R}_r$, we constructed a k-nearest-neighbors (kNN) graph between all chatters in $\mathcal{I}$ to represent their proximity in the feature-space. Finally, we utilized a graph-based label propagation algorithm proposed in [313], seeding nodes (users) with labels as applicable. We tuned parameters of the propagation algorithm empirically to maximize performance.

10.6 Experiments

10.6.1 Baselines

Although no prior works are directly related to the problem we tackle on livestreaming chatbot detection, we adapt certain spam detection approaches which use user similarity and textual features for this setting.

Supervised Spam Classifier (SSC) [20]: We adapt the original work (used for Twitter spam user classification) to our setting. For each user, various features like *max, min, mean, median* of number of words, characters, URLs and IMDs are used to infer in a supervised fashion if user is a chatbot or not. The method works at user-level and does not consider group effects/information at stream level.

SynchroTrap [40]: SynchroTrap is an unsupervised method that operates on user groups; hence, we apply it for each stream to identify constitutent chatbots. We construct edges between any pairs by measuring a soft Jaccard similarity (values are considered similar if they are within small ϵ) between every pair of users. The similarity is computed on two features – (i) IMD, and (ii) number of messages for each user, for every window. We sum the two similarity scores and construct a pairwise similarity graph. We cluster the matrix into two groups via KMeans, and consider the chatbots as the one associated with the group that maximizes performance.

10.6.2 Results on Real Dataset

We evaluate SHERLOCK against the two adapted baselines. We evaluate all three methods at the finest applicable granularity, on their eventual detection performance in detecting chatbots. Stage I is applicable only for SHERLOCK, and we evaluate it's performance using 5-fold cross validation. We discover that SHERLOCK correctly identifies 98.3% of streams, reporting a precision of 0.95. We run Stage II only on those streams that are marked as chatbotted in Stage I; thus, for a misclassified genuine stream, all chatbots are false negatives, and vice versa. For SynchroTrap, we evaluate on all 183 streams in our dataset. Similarly for SSC, we evaluate on all users. We report precision/recall values for each method in their capability to identify chatbots in Table 10.2.

We find that SHERLOCK outperforms both SSC and SynchroTrap in precision and recall, despite SHERLOCK only requiring stream-level labels and SSC requiring much harder to obtain

Table 10.2: Precision and Recall for SHERLOCK, SSC and SynchroTrap on real data.

Model	Genuine Class		Bot Class	
	Precison	Recall	Precision	Recall
SHERLOCK	**97.4%**	**98.6%**	**97.0%**	**94.4%**
SSC	92.6%	96.2%	90.0%	82.8%
SynchroTrap	74.1%	51.8%	35.4%	59.3%

user-level labels. We further conjecture that SSC would perform much worse if the chatbot text was more intelligently generated, while our approach would remain unaffected, due to our text-agnostic feature space. SynchroTrap (unsupervised), works at the stream-level and is unable to leverage information from other streams, hence performing the worst.

10.6.3 Synthetic Dataset Generation

As real world data is not exhaustive, we perform a set of experiments on a variety of synthetic datasets to test the performance of our approach in Stage I/II under unseen, adversarial settings. We consider only our performance, given that SynchroTrap is shown to perform poorly in Table 10.2, and SSC only operates at user-level.

To generate a synthetic, labeled chatbotted livestreaming dataset, we performed the following steps. Firstly, we hired a chatbot service provider and had them attack a dummy stream we had setup ourselves. We avoided targeting others' streams to avoid hurting their reputation. We logged all timestamps relative to the beginning of the stream. We then collected chatlogs with timestamps from a variety of popular, Twitch verified profiles which had high subscriber count. Finally, to generate instances of "chatbotted" streams, we superimposed the original ("legitimate") and synthetic ("botted") chatter, while maintaining respective relative timestamps of both sets of messages. This is a reasonable construction strategy since most chatbots behave independently of legitimate conversation dynamics. We additionally vary control parameters configured through the service provider (the number of chatbots active, N_c, and the maximum delay between consecutive messages d_{max}). By varying these two variables, we construct four attack models:

- **Controlled Chatters (CC)**: We fix N_c and vary d_{max}, mimicking an attack mode where streamers use a constant number of chatbots and tweak delays over the stream.
- **Rapid Increase (RI)**: We start with a small N_c and large d_{max}, and rapidly increase the former and decrease the latter, until the former reaches a certain point. This mimicks streamers trying to poorly emulate organic growth and prolonged engagement.
- **Gradual Increase (GI)**: We consider a similar case as RI, but with longer delays between changing N_c and d_{max}, mimicking a more patient attacker.
- **Organic Growth (OG):** We increase N_c over time, but at each increase, we revert to a large d_{max} before decreasing it (in contrast to keeping d_{max} fixed or a given N_c as in RI/GI), and eventually converging. This mimicks an intelligent attacker, trying to prevent

Table 10.3: F1 score of SHERLOCK across different classification and attack models (Stage I).

Classifier	CC	RI	GI	OG
Decision Tree	0.884	0.943	0.906	0.881
Random Forest	0.889	0.940	0.922	0.899
SVM	0.775	0.711	0.623	0.781
NN	0.842	0.927	0.902	0.892
NN-MLP	0.852	0.925	0.911	0.833
XGBoost	**0.897**	**0.949**	**0.928**	**0.909**

sudden growths in number of chat messages.

To create synthetic datasets, we consider the various (a) attack models $\{$CC, RI, GI, OG$\}$, (b) stream duration $\{0.5, 1, 1.5, 2, 2.5, 3$ hours$\}$, (c) ratio of botted to overall messages and (d) ratio of chatbots to real users $\{40, 60, 80\%\}$. We created multiple simulated attack chatlogs by considering variants of $\{$a,b,c$\}$ and $\{$a,b,d$\}$. This labeled dataset is also used to train the Stage I classifier when classifying unseen streams.

10.6.4 Results on Synthetic Dataset

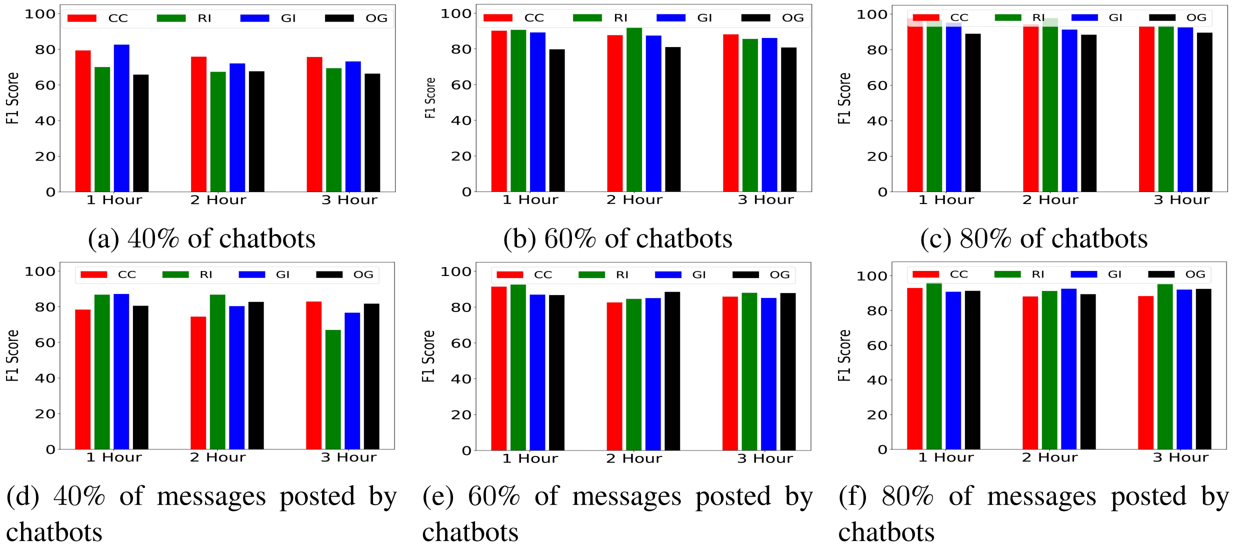
By considering various parameters, we generated 945 CC, 180 RI, 149 GI and 939 OG chatbotted streams. For Stage I, we report performance of SHERLOCK using various traditional supervised learning methods, for different attack models. For Stage II, we consider only streams classified as chatbotted in Stage I. We study the effects of the various synthetic chatlog generation parameters mentioned above.

Stage I: We evaluated performance of different supervised classification models over our feature set, and across varying attack models. We used the corrupted versions of legitimate streams as the positive class, and the original legitimate streams as the negative class. All experiments were conducted using 5-fold cross validation – Table 10.3 shows F1 score for the different classification and attack models.

We found that gradient boosted trees (XGBoost) performed the best amongst the tested methods. Moreover, we discovered that for all classifiers, the CC attack model is the most difficult, while RI is the easiest. We conjecture that this is due to our model's reliance on discriminating IMD features, which are most variant throughout the stream under the CC model (unlike other models, d_{max} never stabilizes in CC).

Stage II: We conduct analysis on all streams marked as botted by the best-performant Stage I classifier. We study the effect of attack model, stream duration, and noise (both ratio of chatbots, and ratio of bot messages). Figure 10.5 shows the collective results in terms of F1 score.

Effect of Attack Model. Unlike in Stage I, we find that the OG model is most challenging. We conjecture that the OG model produces tremendous diversity in the user feature space given many different chatbot configurations, and thus hurts the clustering and propagation steps the most. The GI model proves the easiest to handle; the slow, staggered parameter changes produces several close-by microclusters, which are well-handled by the label propagation.



(a) 40% of chatbots

(b) 60% of chatbots

(c) 80% of chatbots

(d) 40% of messages posted by chatbots

(e) 60% of messages posted by chatbots

(f) 80% of messages posted by chatbots

Figure 10.5: Performance of SHERLOCK on various attack models (bar colors), stream durations (bar groups), noise levels (columns) and noise types (bot users in (a-c), and bot messages in (d-f)). SHERLOCK is robust to noise and performs consistently well across varying adversarial configurations, with F1 scores generally over 0.80.

Effect of Duration. Figures 10.5(a-c) and (d-f) show that duration impacts performance minimally, with slight reduction for higher durations, likely due to increased IMD variety in genuine behaviors.

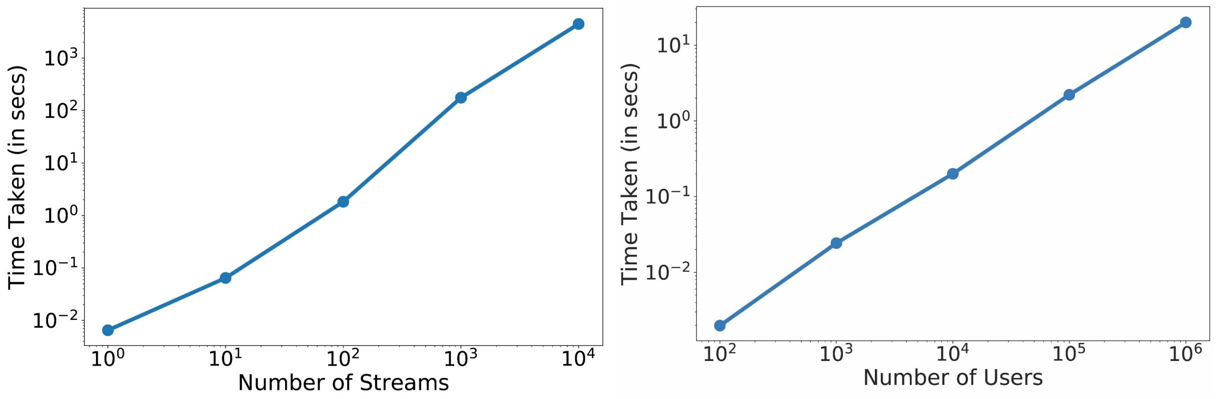
Effect of Noise. We alter between two types of noise models, based on the bot message and bot user ratios. In both cases, increasing the chatbot noise percentage improves performance across various attack models and durations for most configurations. For example, F1 score improves from $78.38(40\%)$, to $91.39(60\%)$, and $92.94(80\%)$ for the 2-hour, CC model, bot user noise setting (red bars in (a-c)). Naturally, higher chatbot signal accentuates the features we use for chatbot seeding and label propagation, lending to better separation.

10.6.5 Scalability

Our two-stage approach is designed to scale naturally, as Stage II (more demanding) works on a significantly reduced set of streams. We evaluate SHERLOCK's scalability in terms of both stages. For Stage I, we generate a synthetic dataset with varying number of streams and show runtime in Figure 10.6(a). For Stage II, we measure time for seeding and propagation; despite $O(kn^2)$ worst-case complexity for k neighbors and n users, Figure 10.6(b) shows near-linear convergence in practice.

10.7 Conclusions

In this work, we tackle the problem of detecting chatbots on livestreaming platforms. Chatbot detection is important due to its direct impact on recommendation, user trust and monetization



(a) Stage I

(b) Stage II

Figure 10.6: SHERLOCK has near-linear runtime in (a) # streams (Stage I) and (b) # users (Stage II).

for these services. We make several contributions in this paper: We are the first to introduce and formalize the chatbot detection problem in the livestreaming setting. Next, we collect and annotate a real-world livestreaming chat dataset from Twitch.tv and compare and contrast genuine and chatbot user behaviors, by identifying key differentiators. We additionally discuss a strategy for obtaining realistic chatlogs with varying attack types and signatures, and employ it in our experimentation. Based on our observations, we propose SHERLOCK, a two-stage approach for detecting chatbotted streams and users with limited supervision. Finally, we evaluate SHERLOCK's effectiveness on both - a real-world dataset (achieving $.97$ precision/recall), and a synthetically generated dataset, showing robustness under various intelligent attack models (achieving 0.80+ F1 score across most settings), and also demonstrate near-linear empirical runtime.

Part V

Anomaly Detection Beyond Social Media

Hemank Lamba, Bryan Hooi, Kijung Shin, Christos Faloutsos. ”zooRank: Ranking Suspicious Entities in Time-Evolving Tensors”. Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), 2017.

CHAPTER 11

INDIVIDUAL METRICS IN GROUP-BASED TEMPORAL FRAUD DETECTION

Most user-based websites such as social networks (Twitter, Facebook) and e-commerce websites (Amazon) have been targets of group fraud (multiple users working together for malicious purposes). How can we better rank malicious entities in such cases of group-fraud? Most of the existing work in group anomaly detection detects lock-step behavior by detecting dense blocks in matrices, and recently, in tensors. However, there is no principled way of scoring the users based on their participation in these dense blocks. In addition, existing methods do not take into account temporal features while detecting dense blocks, which are crucial to uncover bot-like behaviors. In this paper (a) we propose a systematic way of handling temporal information; (b) we give a list of axioms that any individual suspiciousness metric should satisfy; (c) we propose zooRank, an algorithm that finds and ranks suspicious entities (users, targeted products, days, etc.) effectively in real-world datasets. Experimental results on multiple real-world datasets show that zooRank detected and ranked the suspicious entities with high accuracy, while outperforming the baseline approach.

User-based systems, such as web-services like Amazon, Twitter or corporate IT networks, have become popular targets of fraud or attacks. A popular research problem is to detect the spammers/fraudsters/attackers that are trying to attack a given system [22, 124, 131, 259]. Similarly, in the social networks setting, there are multiple websites where anyone can buy fake Facebook page-likes or Twitter followers. Review websites, such as Amazon, Yelp and app-marketplaces, have also been targets for fake reviews. In all these cases, such fraudulent activities take the form of “lockstep” or highly synchronized behavior: such as, multiple users liking the same set of pages on Facebook, or multiple users following the same users almost at the same time on Twitter [22]. Such behavior results in dense blocks in matrices/ tensors. The reason behind these blocks is intuitive, as most of the fraudsters have constrained resources (accounts, IP addresses, time, etc.) and they reuse their resources to add as many fraudulent activities as possible to maximize their profits.

Various methods have been proposed to identify users exhibiting such behavior, which involve finding dense blocks in tensors [124, 259] or clustering in subgraphs [22, 306]. However, for security experts monitoring the systems, it is imperative to know which users are more suspicious than other users, since it directs their attention to such users for further analysis or actions. In this paper we propose a method that ranks entities effectively (see Figure 11.1) for a security analyst to view. Consider Figure 11.2; all three users, A, B and C are participating in dense blocks (as they are part of the 2 rectangles), however their contribution towards the suspiciousness of each block is different. A core question we answer in our paper is as follows:

Informal Problem 1 (Individual Suspiciousness Metric)**.** *Given multimodal temporal data in the form of (userId,productId,. . . ,timestamp), how can we find and score suspicious entities (e.g. users/activities/products/days,etc.)?*

In addition, almost all the social networking websites and services have timestamps associated with every user activity. However, very few approaches in the literature consider temporal features [22]. These timestamps can be useful for detecting fraudsters. However, it is not clear in dense block detection literature, in what ways can we incorporate the temporal information available to us. In this paper we answer the following question:

Informal Problem 2 (Temporal data handling)**.** *Given data in the form of (cat 1, cat 2, . . . , timestamp), how can we generate features from timestamps useful for detecting fraudsters? Here cat 1,cat 2 are any categorical features (generally userId, productId, activityId, ratings, etc.)*

We propose zooRank, a novel approach for successfully scoring entities based on their participation in suspicious dense blocks. We introduce a set of axioms that any ideal individual scoring metric should satisfy. We show theoretically, that our proposed scoring function satisfies the proposed axioms. Additionally, zooRank also provides a framework to make good use of temporal information that generally exists in all the real-world datasets. As shown in Figure 11.1, zooRank successfully finds suspicious users in multiple real-world datasets (Software Marketplace data and Reddit data) with high accuracy. Additionally, the suspicious users found by our method showed clear anomalous patterns. In Figure 11.1(Bottom Left), we see that multiple users are working in groups to target certain products. Similarly, in Figure 11.1(Bottom Right), the suspicious users detected by our method show extremely regular and bot-like behavior resulting in spikes in the inter-arrival time distribution (difference in seconds between consecutive posts).

Our main contributions are as follows:

- **Theory**
  - *Axioms*: We propose a set of axioms that an individual scoring metric for measuring contribution of a user towards a suspicious block should follow.
  - *Metric*: We propose an individual suspiciousness scoring metric.
  - *Proofs*: We further prove that our proposed individual metric follows all the proposed axioms.
- **Temporal Features**: We provide a way of creating temporal features from the timestamp information present in the data.
- **Multimodality and Effectiveness**: The proposed approach zooRank can take into ac-

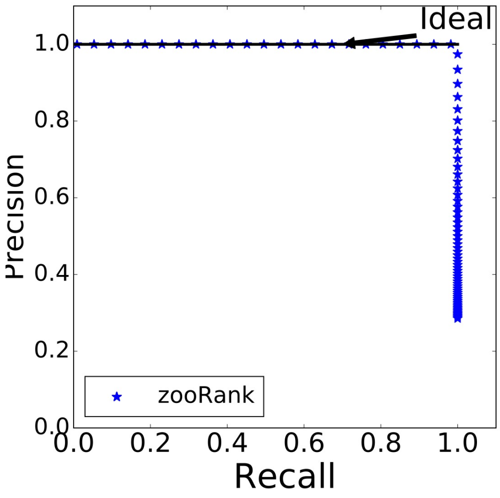
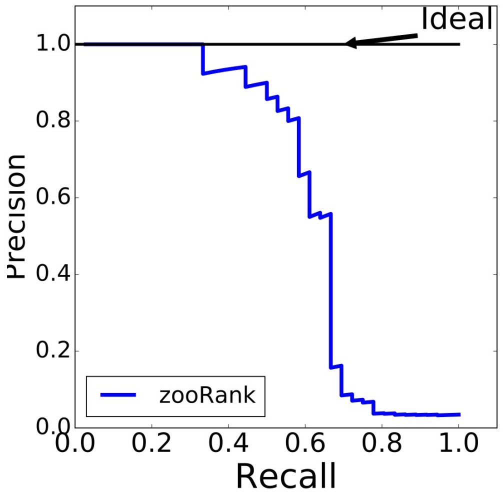
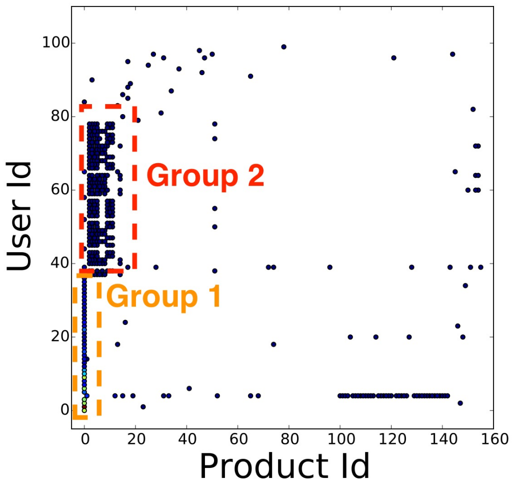
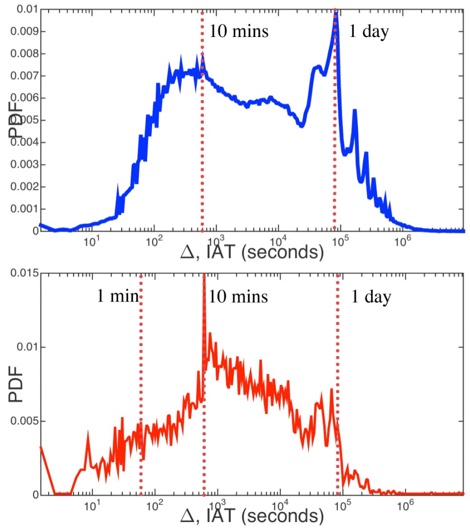
Problem Setup	(User×Product×Rating×Days ×InterArrival Time)	(User × InterArrival Time)
Precision and Recall		
Evidence		

Figure 11.1: **Effectiveness of ZOORANK on real world datasets.** (**Top Left**) Perfect precision-recall on software marketplace dataset. (**Top Right**) ZOORANK obtains good precision recall on Reddit dataset. (**Bottom Left**) Top 100 suspicious users found by ZOORANK show high synchronicity (formed groups) in rating and reviewing top suspicious products. (**Bottom Right**) The suspicious users (bottom; red) detected by ZOORANK for Reddit dataset show irregular spikes in inter-arrival time distribution, as compared to all the users (top; blue).

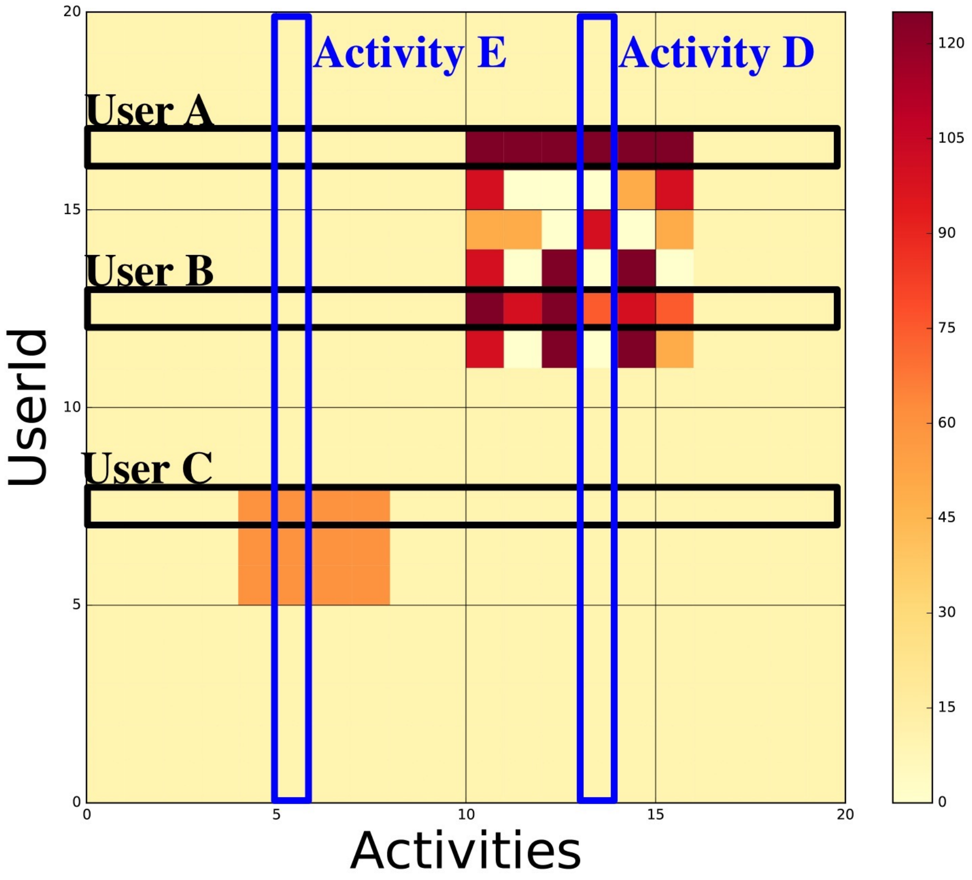


Figure 11.2: How to rank users based on their suspiciousness, matching human intuition (A >B >C) ?

count various features, including temporal features. The approach detects suspicious entities in all modes of the data. We tested zooRank on various real-world datasets and were able to find suspicious entities with high accuracy, revealing interesting fraud patterns.

Reproducability: Our code and link to the datasets used is available at https://goo.gl/2G1DWE

11.1 Related Work

A lot of work exists in the literature which aims at finding dense blocks, but none of the methods present a way of scoring the individual entities in dense blocks. Related work for the given paper comes from the following major subtopics:

Detecting dense blocks: Densest-subgraph identification (i.e., the problem of finding a subgraph with maximum average degree) has been broadly studied in theory, including max-flow based exact algorithms [87] and greedy approximation algorithms [44]. These theoretical results have been extended and applied to anomaly and fraud detection [124, 258] since dense subgraphs (dense blocks) in real-world graph data tend to indicate fraudulent lock-step behavior, such as follower-buying services in Twitter. Spectral methods, which make use of eigen and singular value decomposition, also have been used for detecting dense subgraphs corresponding to 'cut-and-paste' bibliography in patent graphs [233], lock-step followers [131] and small-scale stealthy attacks [253] in social networks. Other approaches for dense-subgraph detection include co-clustering [22] and belief propagation [223]. Recently, dense-block detection in multi-aspect data also has been researched [132, 259] for spotting groups synchronized in multiple aspects, such as IPs, review scores and review keywords. For our experiments, we use the best performing dense subgraph detection method M-Zoom [259]. The existing methods, however aim at only finding blocks, and do not provide a rank-list of users to inspect according to their suspiciousness.

Scoring Anomalies: Evaluating the anomalousness or suspiciousness of individuals is complementary to detecting dense blocks, which correspond to group activities. A widely-used approach is to detect outliers, i.e., observations that deviate greatly from other observations. Outlier detection methods are divided into parametric methods assuming underlying data distribution [19, 112] and non-parametric methods using local features, such as distances to neighbors [145] and local density [35, 169]. For graph data, on the other hand, various approaches, based on minimum description length [65, 102], neighborhood information [270], egonet features [11] have been proposed for scoring nodes. Many methods do exist in the literature, which use temporal information such as inter-arrival time [70, 288]. These features have been used to successfully detect bot-like behavior [70].

Our proposed method zooRank scores each entity (*individual-scoring*) in any of the dimensions (*multimodal*) of the tensor based on the entity's participation in the suspicious dense blocks (*dense-blocks*). It provides ways of transforming temporal data into useful features and thus handles both *numerical and categorical features*.

A comparison between zooRank and other algorithms is summarized in Table 11.1. Our proposed method zooRank is the only one that matches all specifications.

	N-dim Outlier Methods			Point Processes		Graph/Tensor based Methods					
	GFADD /FADD [169]	**LoF** [35]	**Robust Random Cut**[102]	**RSC** [70]	**Self Feeding** [288]	**SPOKen** [233]	**CopyCatch** [22]	**CrossSpot** [132]	**M-Zoom** [259]	**FRAUDAR** [124]	**ZOORANK**
Dense Blocks						✓	✓	✓	✓	✓	✓
Individual Scoring	✓	✓	✓								✓
Numerical & Categorical Features							✓				✓
Multimodal and Extensible								✓	✓		✓
Temporal Features			✓	✓	✓		✓				✓

Table 11.1: Comparison of other methods and their features

11.2 Preliminaries and Problem Definition

11.2.1 Problem Definition

Definition 1 (K-way timed tensor)**.** *A K-way timed tensor is a higher-order matrix containing entries of the form (category 1, category 2, ...,category K, timestamp).*

Many types of data including "like" data from Facebook (UserId, PageId, Timestamp), "follow" data from Twitter (UserId, FolloweeId, Timestamp), activity log from an organization (UserId, OperationId, Timestamp) or network data (Source IP, Source Port, Destination IP, Destination Port, Timestamp) all can be formulated as a K-way timed tensor. We now give a precise definition of the problem statements.

Problem 1 (Temporal Features Handling)**.** *Given a K-way timed tensor $\mathcal{A}$, how can we effectively transform the temporal features associated with $\mathcal{A}$ to generate a categorical tensor $\mathcal{X}$?*

Problem 2 (Individual-Suspiciousness)**.** *Given a L-way categorical tensor $\mathcal{X}$ of size $N_1 \times N_2 \times \cdots \times N_L$ with non-negative entries, compute a* ***score function*** *$f_{\mathcal{X}}(i)$, which defines the suspiciousness of entity i in the $m(i)^{th}$ mode of $\mathcal{X}$ with respect to the overall tensor $\mathcal{X}$.*

11.2.2 Block Level Suspiciousness Metrics

In this paper, we consider three block-level suspiciousness metrics although our proposed method is not restricted to them. The metrics are Arithmetic (g_{ari}), Geometric (g_{geom}) and Density (g_{susp}). Arithmetic computes the arithmetic average mass of a sub-block $\mathcal{Y}$ of a tensor $\mathcal{X}$. Similarly, Geometric metric is the geometric average mass of the block. The Density metric is the KL-divergence (Kullback Leibler) between the distribution of the mass in the sub-block with respect to the distribution of the mass in the tensor. These metrics are explained in the following sections.

Symbol	Definition
$\mathcal{X}$	*Input categorical L-way tensor*
$\mathcal{Y}$	*Dense block within tensor $\mathcal{X}$*
$N_{\mathcal{Y}}^{i}$	*Size of ith mode of block $\mathcal{Y}$*
$m(i)$	*Mode of entity i*
$\rho_{\mathcal{Y}}$	*Density of block $\mathcal{Y}$*
$C_{\mathcal{X}}$	*Sum of the entries in $\mathcal{X}$*
$C_{\mathcal{Y}}$	*Sum of the entries in $\mathcal{Y}$*
$C_{\mathcal{Y}}(i)$	*Mass of entity i in $\mathcal{Y}$*
$V_{\mathcal{Y}}$	*Volume of the block $\mathcal{Y}$*
$g()$	*Block suspiciousness scoring function*
$f()$	*Individual-Suspiciousness scoring function*
$\delta_{\mathcal{Y}}(i)$	*Block level suspiciousness of entity i in block $\mathcal{Y}$*
$\mathcal{B}$	*List of suspicious dense blocks*
M	*Number of suspicious blocks to be considered*

Table 11.2: Symbols and Definitions

11.2.3 Axioms

In this sub-section, we establish axioms that a good score function $f = f_{\mathcal{X}}(i)$ should satisfy. The suspiciousness of an entity should be based on its participation in dense blocks $\mathcal{B}$. Hence, our first two axioms govern the scores with respect to a single block $\mathcal{Y} \in \mathcal{B}$: our third axiom then governs how the single-block scores are combined to form $f_{\mathcal{X}}(i)$.

Let $\rho_{\mathcal{Y}}$ be the density (i.e. mass divided by volume) of $\mathcal{Y}$, and $\rho_{\mathcal{Y}}(i)$ be the density of the slice of $\mathcal{Y}$ defined by entity i. Similarly, let $\mathcal{C}_{\mathcal{Y}}(i)$ denote the mass of that same slice. The entire list of symbols is shown in Table 11.2.

Axiom 1 (Mass)**.** *If an entity a has more mass than entity b in a block and given the fixed size of block in both the modes $m(a)$ and $m(b)$, then entity a is more suspicious. Formally*

$$\begin{aligned} \textit{IF } \mathcal{C}_{\mathcal{Y}}(a) > \mathcal{C}_{\mathcal{Y}}(b), \quad \textit{AND} \quad \mathcal{N}_{\mathcal{Y}}^{m(a)} = \mathcal{N}_{\mathcal{Y}}^{m(b)}, \\ \textit{THEN, } \delta_{\mathcal{Y}}(a) > \delta_{\mathcal{Y}}(b) \end{aligned}$$

This is represented in Figure 11.2, how entities are ranked by suspicion in the top right block (User A > User B > Activity D).

Axiom 2 (Concentration)**.** *Given two entities a, b in different modes $m(a), m(b)$, where number of entities in one mode ($N_{\mathcal{Y}}^{(m(a))}$) is less than the number of entities in the second mode ($N_{\mathcal{Y}}^{(m(b))}$), then for fixed density, entity a is more suspicious than entity b.*

Formally,

$$\begin{aligned} \textit{IF } N_{\mathcal{Y}}^{m(a)} < N_{\mathcal{Y}}^{m(b)}, \quad \textit{And} \quad \rho_{\mathcal{Y}} = \rho_{\mathcal{Y}}(a) = \rho_{\mathcal{Y}}(b) \\ \textit{THEN,} \quad \delta_{\mathcal{Y}}(a) > \delta_{\mathcal{Y}}(b) \end{aligned}$$

This is represented in Figure 11.2, consider the lower left block where (User C > Activity E).

Axiom 3 (Monotonocity). *If for every block, entity a has higher suspiciousness than entity b, then entity a has higher overall suspiciousness.*

Formally,

$$\begin{aligned} \textit{IF } & \delta_{\mathcal{Y}}(a) > \delta_{\mathcal{Y}}(b) \quad \forall \mathcal{Y} \in \mathcal{B} \\ \textit{THEN } & f_{\mathcal{X}}(a) > f_{\mathcal{X}}(b) \end{aligned}$$

11.2.4 Shortcomings of Other Metrics

While these axioms are simple and intuitive, many other candidate metrics are not able to satisfy them. We consider some of them, and show why they fail.

Block Score: One simple metric to consider is the block suspiciousness score itself. The metric is to assign each individual the maximum block suspiciousness score out of all the blocks it is part of. The metric doesn't change if the two entities have different contributions to the block, and hence fails Axiom 1 (*Mass*) and Axiom 2 (*Concentration*).

SVD-score: Any matrix $\mathbf{A}$ can be decomposed using SVD decomposition as follows: $\mathbf{A} = \mathbf{U\Sigma V^T}$. Each of the singular values in $\mathbf{\Sigma}$ represents the singular value related to a dense block that exists in the dataset. The metric here is the score of the maximum component for each user. This metric would again fail Axiom 1 (*Mass*) and Axiom 2 (*Concentration*).

Average δ-Block Score: Another proposed metric could be the average of all the contributions by the given entity to all the suspicious blocks. The contribution to a block is computed as the difference in the suspiciousness between the block and the block after removing the specified entity. This statistic fails to satisfy Axiom 3 (*Monotonocity*) as if entity 1 has higher suspiciousness in 2 blocks than entity 2, but entity 2 exists only in one of the blocks, then the mean statistic is ambiguous.

From the above section, we can see that the metrics based on just the block statistics and the basic metrics on the aggregation of the block statistics do not work. In the following section, we present our approach.

11.3 Proposed Approach:zooRank

11.3.1 Temporal Feature Handling

As mentioned, any data from a social networking website or a web service can be represented as a K-way timed tensor. We propose a way to handle such tensors by converting the numerical timestamp mode into interpretable categorical features. We propose to generate 0^{th}-order, 1^{st}-order, and temporal folding features.

- 0^{th}**-order features:** The 0^{th} order features bucketize the timestamp into number of days, hours, minutes, etc. passed since the first observation was made. The temporal resolution can be chosen by practitioners based on the typical level of temporal variation present in their dataset.
- 1^{st}**-order features:** Inter-arrival time is defined as the time interval between 2 consecutive timestamps of the same user. [70] found that bots tend to display regular inter-arrival

time behavior such as performing an activity every exactly 5 minutes, due to automated scripts. To capture this pattern, we propose 1^{st}-order features, which is the log-bucketized inter-arrival time between 2 consecutive operations of a user (generalizable to any entity).

- **Temporal folding features:** We propose another way to detect fraudsters showing periodic behavior, which are common in bot-like behavior. For instance, a group of anomalous users might try to perform multiple login activities only from Wednesday 10 PM to 11 PM, or only on a specific day of the week. We work with 3 such features: 1) day of the week, 2) hour of the day and 3) hour of the week. We call these features temporal folding features.

11.3.2 Proposed Metric

Our metric is based on the δ-contribution of each entity towards the block suspiciousness score. We first define the δ-contribution for a given entity i in mode $m(i)$ of a specific block $\mathcal{Y} \in \mathcal{B}$, where $\mathcal{B}$ is a list of blocks. We denote this by $\delta_{\mathcal{Y}}(i)$

Definition 2 (Entity's Block-level Suspiciousness ($\delta_{\mathcal{Y}}(\mathbf{i})$))**.** *We define $\delta_{\mathcal{Y}}(i)$ as the difference between the suspiciousness score of block $\mathcal{Y}$ and block $\mathcal{Y}$ after removing entity i from the block i.e., $\delta_{\mathcal{Y}}(i) = g(\mathcal{Y}) - g(\mathcal{Y} \setminus i)$*

We need to aggregate the δ-metric over the entire list of blocks $\mathcal{B}$, in such a way that the given axioms are satisfied. We propose two metrics both of which satisfy the given axioms. The first metric is the sum of the δ-contributions, and the second is the maximum of the δ-contributions. We define the maximum metric as follows:

$$f_{\mathcal{X}}(i) = \max_{\mathcal{Y} \in \mathcal{B}}(\delta_{\mathcal{Y}}(i))$$

We empirically found that the maximum metric performs the best on the real-world datasets, and hence for the rest of the paper, all references to the proposed metric is for the maximum version of the metric.

11.3.3 Algorithm

After handling the temporal features, we produce a categorical tensor $\mathcal{X}$. Algorithm 3 defines the outline of zooRank. The first step is to compute suspicious blocks for the given tensor $\mathcal{X}$. To compute suspicious blocks, any existing method for block detection can be used.

We first find the M top suspicious dense blocks as determined by g (Line 1), where g is one of the metrics defined in Section 11.2.2. These top M suspicious blocks are stored in the list $\mathcal{B}$. For every entity i that has occurred at least once in any of the blocks in $\mathcal{B}$, we compute the individual suspiciousness score function f. This score function captures the contribution of a particular entity towards making the block suspicious. To do this, we compute the marginal contribution of each node towards that block. This is equivalent to removing the entity i from the block, and re-computing the suspiciousness score (Lines 6-7). The difference between the new suspiciousness score and the original suspiciousness score is the marginal contribution of entity i. We compute the marginal contribution of each entity i over all the blocks (Lines 4-8). We define the individual suspiciousness score of the entity i as the maximum of the marginal

Algorithm 3 zooRank: Individual Suspiciousness Detection

```
Require: Tensor $\mathcal{X}$, block scoring function $g$, number of blocks to consider $M$, mode $j$ to consider
Ensure: Individual scores for each entity $i$ over the entire tensor: $f_{\mathcal{X}}(i)$
1: $\mathcal{B}$ = ComputeDenseBlocks($\mathcal{X}, M, g$)
2: for each entity $i \in N_j$ do
3:     $\delta_i = []$
4:     for $\mathcal{Y} \in \mathcal{B}$ do
5:         if $i \in \mathcal{Y}$ then
6:             Create new block $\mathcal{Y}'$ by removing entries of entity $i$
7:             Append $(g(\mathcal{Y}) - g(\mathcal{Y}'))$ to $\delta_i$
8:     $f_{\mathcal{X}}(i) = \max(\delta_i)$
9: Sort and output $f_{\mathcal{X}}(i)$
```

contributions of entity i (Line 9). Another potential metric is to replace the maximization in Line 9 by the sum function. We conduct experiments with that metric as well.

This formulation of the scores $f_{\mathcal{X}}(i)$ satisfies intuitively reasonable properties, namely our axioms defined in Section 11.2.3:

Theorem 1. *The scores $f_{\mathcal{X}}(i)$ computed by Algorithm 3, using any of the metrics g_{ari}, g_{geo}, or g_{susp}, satisfies Axioms 1 to 3.*

Proof. :We first start by defining some of the standard block suspiciousness methods as follows:

$$g_{ari}(\mathcal{Y}, \mathcal{X}) = C_{\mathcal{Y}} / (\sum_j N_{\mathcal{Y}}^j / L)$$
$$g_{geo}(\mathcal{Y}, \mathcal{X}) = C_{\mathcal{Y}} / (V_{\mathcal{Y}}^{1/L})$$
$$g_{susp}(\mathcal{Y}, \mathcal{X}) = V_{\mathcal{Y}} \cdot \mathrm{D}(\rho_{\mathcal{Y}} || \rho_{\mathcal{X}})$$

where $\mathrm{D}(\rho_{\mathcal{Y}} || \rho_{\mathcal{X}}) = \rho_{\mathcal{X}} - \rho_{\mathcal{Y}} + \rho_{\mathcal{Y}} \log \frac{\rho_{\mathcal{Y}}}{\rho_{\mathcal{X}}}$.

zooRank satisfies Axiom 1 (Mass)
If we fix the block's dimensions $N_{\mathcal{Y}}^1, \ldots, N_{\mathcal{Y}}^L$, all 3 metrics above are strictly increasing in the mass of the block (i.e. $C_{\mathcal{Y}}$); this can be inferred directly from the form of g_{ari} and g_{geo}, and for g_{susp}.

As $\mathcal{C}_{\mathcal{Y}}(a) > \mathcal{C}_{\mathcal{Y}}(b)$, thus $\mathcal{Y} \setminus a$ has lower mass than $\mathcal{Y} \setminus b$, and since g is strictly increasing in mass (for fixed block dimensions), we get $g(\mathcal{Y} \setminus a) < g(\mathcal{Y} \setminus b)$. Therefore:

$$\begin{aligned}\delta_{\mathcal{Y}}(a) &= g(\mathcal{Y}) - g(\mathcal{Y} \setminus a) > g(\mathcal{Y}) - g(\mathcal{Y} \setminus b)\\ &= \delta_{\mathcal{Y}}(b)\end{aligned}$$

zooRank satisfies Axiom 2 (Concentration)
Using the same reasoning as above, it suffices to show $g(\mathcal{Y} \setminus a) < g(\mathcal{Y} \setminus b)$. Note that $N_{\mathcal{Y}}^{m(a)} < N_{\mathcal{Y}}^{m(b)} \Rightarrow V_{\mathcal{Y} \setminus a} < V_{\mathcal{Y} \setminus b}$ (since removing from a smaller mode decreases the volume more). Consider each metric g_{ari}, g_{geo}, and g_{susp} separately:

- **case 1:** g_{ari}.
  Here $\mathcal{Y} \setminus a$ and $\mathcal{Y} \setminus b$ have the same sum of block dimensions, and $\mathcal{C}_{\mathcal{Y}\setminus a} = \rho_{\mathcal{Y}} \cdot \mathcal{V}_{\mathcal{Y}\setminus a} < \rho_{\mathcal{Y}} \cdot \mathcal{V}_{\mathcal{Y}\setminus b} = \mathcal{C}_{\mathcal{Y}\setminus b}$ so that $g_{ari}(\mathcal{Y} \setminus a) < g_{ari}(\mathcal{Y} \setminus b)$.
- **case 2:** g_{geo}.
  Note that $g_{geo}(\mathcal{Y}) = C_{\mathcal{Y}}/(V_{\mathcal{Y}}^{1/L}) = \rho_{\mathcal{Y}} \cdot V_{\mathcal{Y}}/(V_{\mathcal{Y}}^{1/L}) = \rho_{\mathcal{Y}} \cdot V_{\mathcal{Y}}^{\frac{L-1}{L}}$. Thus:

$$g_{geo}(\mathcal{Y} \setminus a) = \rho_{\mathcal{Y}} \cdot (\mathcal{V}_{\mathcal{Y}\setminus a})^{\frac{L-1}{L}} < \rho_{\mathcal{Y}} \cdot (\mathcal{V}_{\mathcal{Y}\setminus b})^{\frac{L-1}{L}} = g_{geo}(\mathcal{Y} \setminus b)$$

- **case 3:** g_{susp}.
  $g_{susp}(\mathcal{Y} \setminus a) = V_{\mathcal{Y}\setminus a} \cdot \mathrm{D}(\rho_{\mathcal{Y}} || \rho_{\mathcal{X}}) < V_{\mathcal{Y}\setminus b} \cdot \mathrm{D}(\rho_{\mathcal{Y}} || \rho_{\mathcal{X}}) = g_{susp}(\mathcal{Y} \setminus b)$

ZOORANK satisfies Axiom 3 (Monotonocity)

$$f_{\mathcal{X}}(a) = \max_{\mathcal{Y} \in \mathcal{B}} \delta_{\mathcal{Y}}(a) > \max_{\mathcal{Y} \in \mathcal{B}} \delta_{\mathcal{Y}}(b) = f_{\mathcal{X}}(b).$$

□

11.4 Experiments

In this section, we conducted experiments to answer the following questions:

- **Q1**: How effectively does ZOORANK find suspicious entities across all modes?
- **Q2**: How generalizable is ZOORANK over different datasets?
- **Q3**: Does ZOORANK scale linearly with size of the data ?

11.4.1 Datasets

We used various real-world datasets including a software marketplace dataset, a dataset from a popular social news aggregation website (Reddit), a dataset about Indian elections from Twitter, and a research lab's intrusion detection dataset.

- **Software Marketplace Dataset (SWM)**: We used the SWM dataset that was used previously by [12]. The dataset contains the reviews for all the products (software) under the entertainment category of the marketplace. The dataset contains 1,132,373 reviews from 966,839 unique users for 15,094 products. Each review has a rating from 1 to 5, and the timestamp on which the review was posted. The dataset, thus is in the format (UserId, ProductId, Rating, Timestamp). Previous studies [12, 306] manually annotated ground truth labels for suspicious users, which we considered as our ground truth.
- **Reddit Dataset**: Reddit is a social news aggregator website, which allows users to post, comment on, upvote and downvote stories. The dataset was collected and analyzed by [70]. The dataset contains 1,020,834 user comments for 1,036 users. The Reddit dataset is in the form (UserId, #Upvotes, #Downvotes, Length, Timestamp). The dataset has information about ground truth suspicious user accounts.

- **DARPA Intrusion Detection**: The DARPA intrusion detection dataset contains a sample of network data for the US Air Force laboratory[1]. The dataset contains records in the format (Source IP, Destination IP, Timestamp). Further, it also contains labels for anomalous connections. For ground truth, we considered any source IP address that participates in at least 10 such anomalous connections, and any destination IP address that participates in at least 400 such connections. We altered this definition for ground truth thresholds and still achieved similar results as mentioned in the paper.
- **Indian Elections 2014 Dataset**: We collected tweets from 2014 Indian Elections. We crawled all the tweets from the 10% Sample API (Decahose). All the tweets contain the top 5 hashtags on Indian Elections per week. We further considered only those users who have at least 2 tweets in our dataset. This led us to a dataset of tweets from March, 2014 consisting of 10,786 users.
- **Simulated Dataset**: We also tested our approach on a simulated dataset. For simulation, we used a realistic way of generating user-timestamps [70], then for each of the timestamp, we added activities based on a Poisson distribution. We simulated 3 blocks, comprising of 300, 400 and 200 genuine users respectively, where each block has different parameters for the activity Poisson distribution. For the suspicious blocks, we simulated three blocks for 50, 25 and 25 users respectively. The first block does the most popular activity over the entire duration of the simulation and with random inter-arrival times. The second and third block do the second most and third most popular activities at a steady inter-arrival time of 1 minute on a single day.

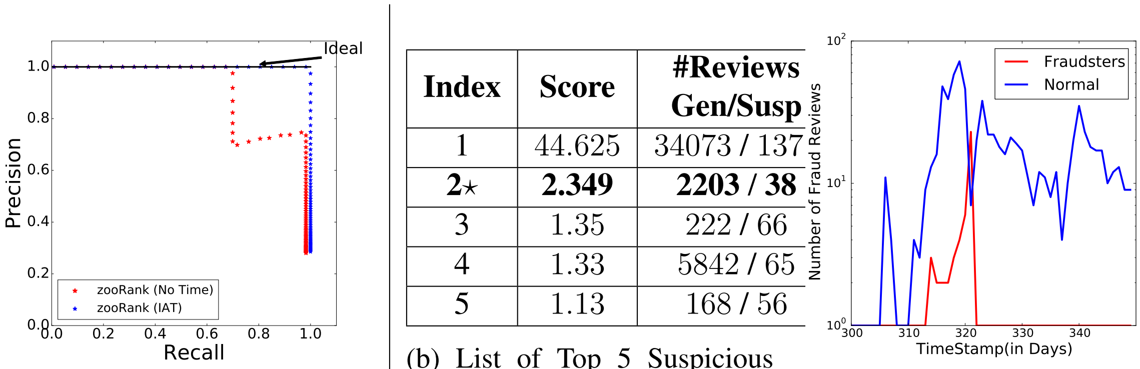
Experimental Settings: All our experiments were conducted on a machine on Intel(R) Xeon(R) CPU W3530 @ 2.80 GHz and 24 GB RAM. For all our experiments, we chose $M = 30$ and used M-Zoom [259] for dense block detection. We created multiple tensors based on different resolutions of time features (such as day of week, hour of the day, Inter-arrival time (in seconds, bucketized), etc.). However, we reported only the best accuracy obtained. The choice of what tensor to use, what block-level metric to use, and what value of M is appropriate, is for the practitioner to decide and depends on the type of data, on which the method is being applied.

11.4.2 Q1. Effectiveness of zooRank

To test the effectiveness of zooRank, we compare our ranking of the suspicious entities with the ground truth suspicious users in our datasets. We further test the accuracy of our method on the SWM dataset. For software marketplace, we experimented with different versions of temporal features. Note that our algorithm achieves 100% accuracy in identifying suspicious users in the SWM dataset. From Figure 11.3a, we observed that adding the inter-arrival time feature increased the accuracy of the method. Our algorithm can rank entities in multiple modes; hence, we also tried to rank the products on basis of their suspiciousness. Though we do not have ground truth for which products were suspicious, we analyzed the top 5 suspicious products in Table 11.3b. We used the number of reviews by ground truth fraudsters as an indicator for suspiciousness. It can be observed that all the suspicious products are popular (high number of total reviews) and have also been targeted significantly from fraudsters (high number of fraud

[1] https://www.ll.mit.edu/ideval/data/

users). We also noticed that most of the reviews by fraudsters were highly synchronized and a large majority came on a single day (Figure 11.3c).



Index	Score	#Reviews Gen/Susp
1	44.625	34073 / 137
2⋆	**2.349**	**2203 / 38**
3	1.35	222 / 66
4	1.33	5842 / 65
5	1.13	168 / 56

(a) Effect of inter-arrival time: Performance on the user mode.

(b) List of Top 5 Suspicious Products.

(c) Performance on Product mode of SWM dataset.

Figure 11.3: **ZOORANK is effective.** (a) It gives nearly 100% accuracy while identifying suspicious users in the SWM dataset. (b) ZOORANK marks products reviewed by known fraudsters as suspicious. (c) Product #2 received nearly all of it's reviews by fraud users on one single day.

11.4.3 Q2. Generalizability of ZOORANK

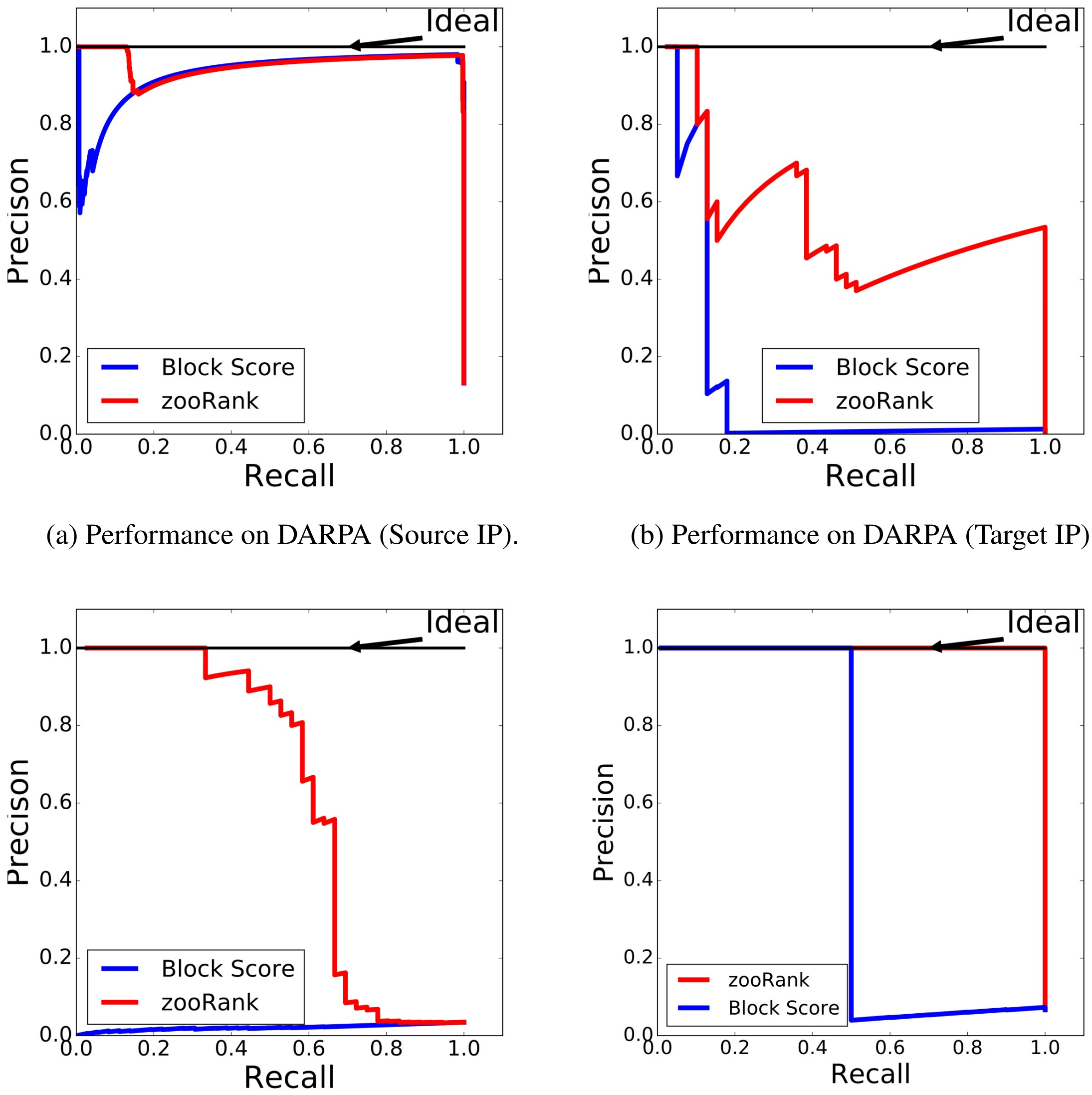
We tested our method on multiple real-world datasets. In Table 11.3, we present our accuracy on each dataset. We observed that using maximum of the marginal contributions is better than using sum for all of the cases. Further, we also compared our method with a baseline approach. We define the following baseline:

Block Score: defined as the maximum of all block suspiciousness scores a block is part of. From Figure 11.4, it can be observed that our approach clearly is better than the mentioned approach.

Dataset	F1-Score(SUM)	F1-Score(MAX)	Tensor
Reddit	0.62	**0.67**	User × Inter-Arrival Time (IAT)
SWM	0.98	**1.0**	User × Product × Rating ×Day × IAT
DARPA (SrcIP Mode)	0.97	**0.988**	SrcIP × DstIP × Day × IAT
DARPA (DstIP Mode)	0.29	**0.37**	SrcIP × DstIP × Hour × IAT

Table 11.3: ZOORANK is generalizable over multiple datasets, and multiple modes that exist in the dataset.

For Indian elections data, we did not have any ground truth. We extracted the top 100 suspicious users and evaluated them manually. The results for top 100 suspicious users are shown in Figure 11.5. The user ids are sorted by their suspiciousness score, and plotted on the scatter plot



(a) Performance on DARPA (Source IP).

(b) Performance on DARPA (Target IP).

(c) Performance on Reddit Dataset.

(d) Performance on the Simulated Dataset

Figure 11.4: **zooRank is generalizable**. zooRank outperforms the baseline across different modes (see (a) and (b)) and across multiple datasets (see (c) and (d))

along with top 100 suspicious hashtags. Figure 11.5 clearly shows groups of suspicious users. It is evident that the first two users are "hashtag hijackers". These two users tweeted spam messages with other hashtags but also focussed on generic hashtags related to the Indian elections. Both of these users have an identical behavior, which imply they do follow "lock-step" behavior. The second group of users were tweeting hashtags related to themselves and also generic hashtags related to the elections ("self-promoters"). We also spotted the user who tweets out all the trending topics at regular intervals, possibly through automated scripts ("trending topic aggregator"). We believe that the remaining users are users who were discussing indian elections a lot and were influencers in the political discussion. On further analysis, 20 users out of the 100 users were already suspended by Twitter. Thus, our algorithm was able to identify users that were considered spam by Twitter but also users that were missed by Twitter algorithm ("self-promoters") but were clearly malicious.

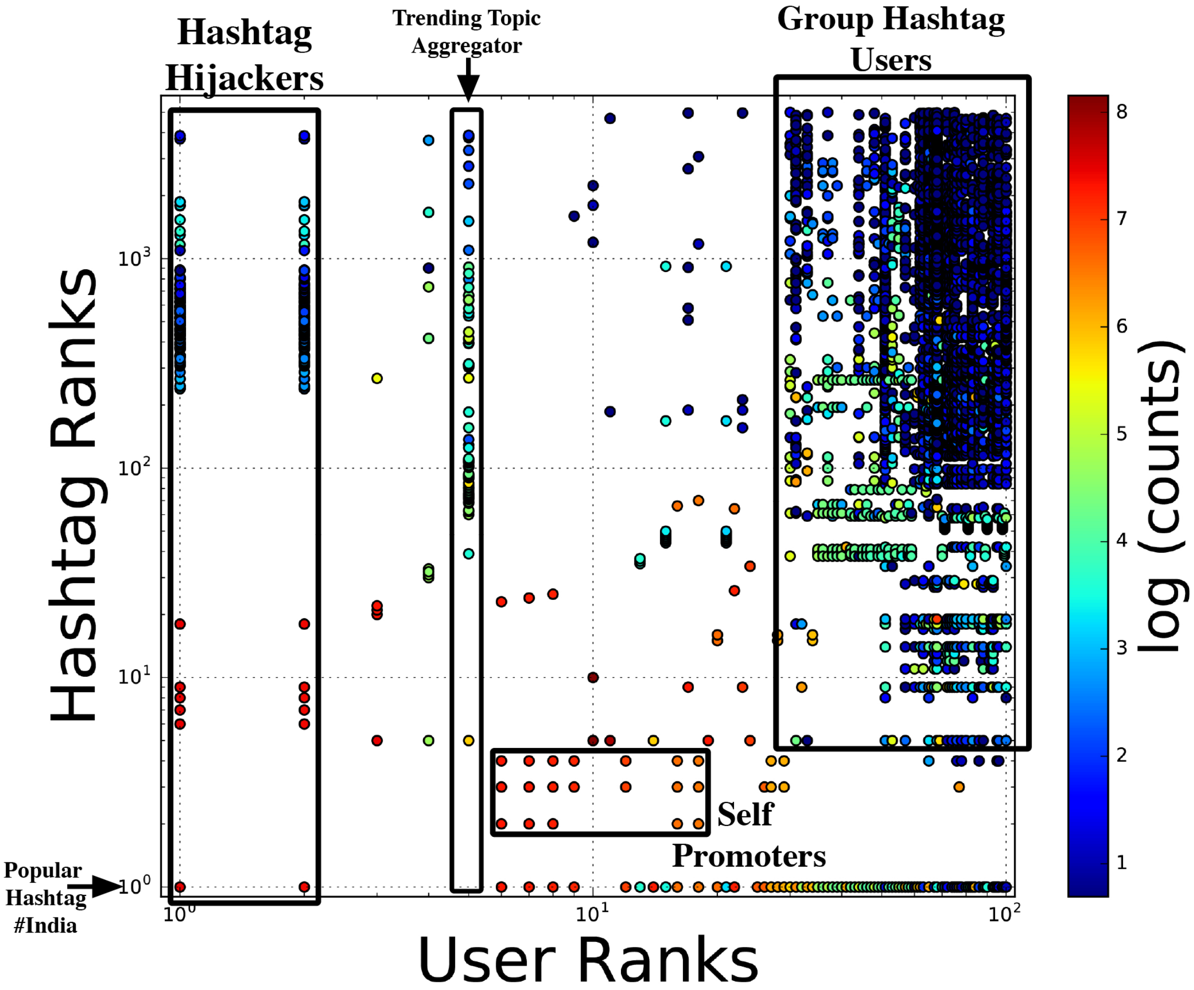
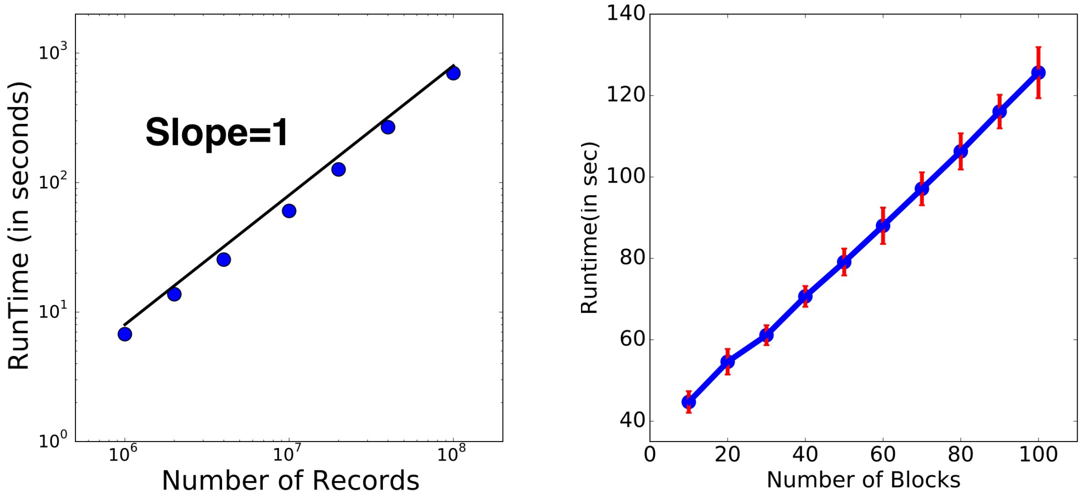


Figure 11.5: **ZOORANK identifies fraudulent suspicious behavior in Twitter**: Top 100 suspicious users, and top 100 products as identified by ZOORANK. We can notice clearly the groups of suspicious users.

11.4.4 Q3. Scalability of zooRank

In this section, we evaluate the scalability of the zooRank. We measure the effect of number of blocks, number of entries and effect of the density metric on the runtime of zooRank. To study the effect of number of tuples, we generated the dataset with given number of entries in 3 dimensions, where cardinality of each dimension is 10^6. For all our results, we used arithmetic metric and operated on the most suspicious 30 blocks. The results are shown in Figure 11.6a, showing that our method scales linearly both in the data size and the number of blocks searched for. For the effect of number of blocks, we generated a dataset with 10^4 records with the similar number of entries in each dimension, and arithmetic suspiciousness metric was used.



(a) Number of Tuples.

(b) Number of Blocks.

Figure 11.6: **Scalability of zooRank** (a) zooRank scales linearly with number of records. (b) zooRank scales linearly with number of blocks we want to find.

11.5 Conclusions

In this paper, we proposed a set of axioms that a given individual suspiciousness scoring metric should follow. We presented such a metric that satisfies all the proposed axioms. Specifically, our contributions are as follows:

- **Individual-Suspiciousness Metric**: We propose a suspiciousness metric which scores each entity participating in dense blocks. The proposed criteria $f_{\mathcal{X}}(i)$ satisfies intuitive axioms.
- **Temporal Features**: The proposed method provides ways to transform the numerical timestamp mode to information rich categorical temporal features.
- **Effectiveness**: The proposed method zooRank was successfully tested on various real-world datasets. It scored the suspicious entities with high accuracy, and also uncovered

interesting fraud patterns.

- **Scalability**: The method is linearly scalable with the size of the data and can be used for *big-data* problems (see Figure 11.6).

Hemank Lamba, Leman Akoglu. ”Learning On-the-Job to Re-rank Anomalies from Top-1 Feedback”. Proceedings of the 2019 SIAM International Conference on Data Mining (SDM), 2019.

CHAPTER 12

INCORPORATING HUMAN FEEDBACK FOR ANOMALY DETECTION

In many anomaly mining scenarios, a human expert verifies the anomaly at-the-top (as ranked by an anomaly detector) before they move on to the next. This verification produces a label—true positive (TP) or false positive (FP). In this work, we show how to leverage this label feedback for the top-1 instance to quickly re-rank the anomalies in an online fashion. In contrast to a detector that ranks once and goes offline, we propose a detector called OJRANK that works alongside the human and continues to learn (how to rank) *on-the-job*, i.e., from every feedback. The benefits OJRANK provides are two-fold; it *reduces* (i) *the false positive rate* by 'muting' the anomalies similar to FP instances; as well as (ii) *the expert effort* by elevating to the top the anomalies similar to a TP instance. We show that OJRANK achieves statistically significant improvement on both detection precision and human effort over the offline detector as well as existing state-of-the-art ranking strategies, while keeping the per feedback response time (to re-rank) well below a second.

Given an anomaly mining setting in which a human expert needs to verify the anomalousness of instances as ranked by a detection algorithm, starting at the top of the ranked list, how can we leverage the labels they produce along the way to re-rank the anomalies? How can we update the ranking fast, without stalling the expert?

Anomaly detection has been mainly considered a stand-alone task that precedes any action-taking. In most applications, however, post-hoc human validation is either mandatory or necessary. For example, in auditing systems for insurance claims, expense invoices, tax returns, etc., the anomalies may be indicative of errors or fraud. However, one cannot automatically decline to pay-back the anomalous cases— errors must be located and fraudulent activities must be verified by human experts. For surveillance systems such as user behavior tracking or systems monitoring, anomalies may be indicative of malicious activities, however it may be undesirable to automatically shut down the anomalous user accounts or the running processes before verification. Similarly for knowledge discovery tasks, such as spotting new objects in sky images or

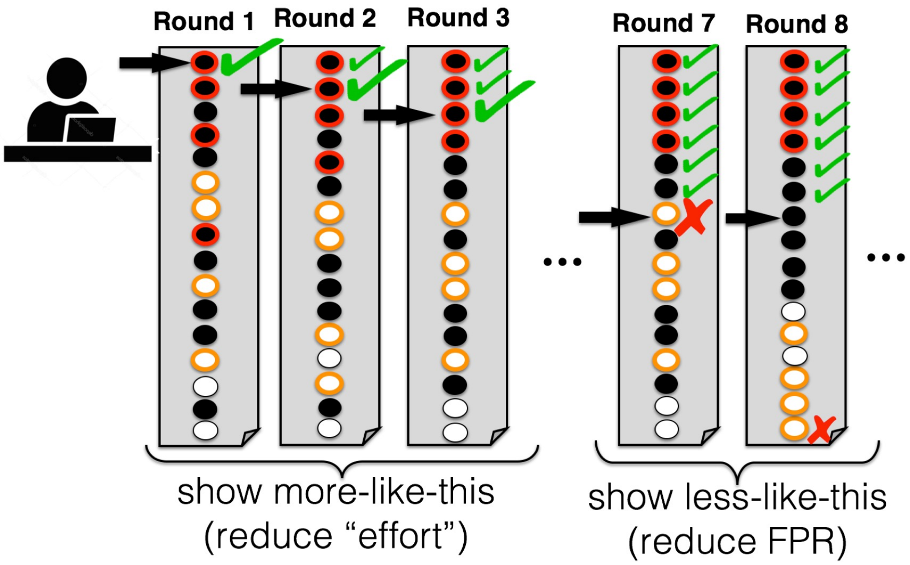


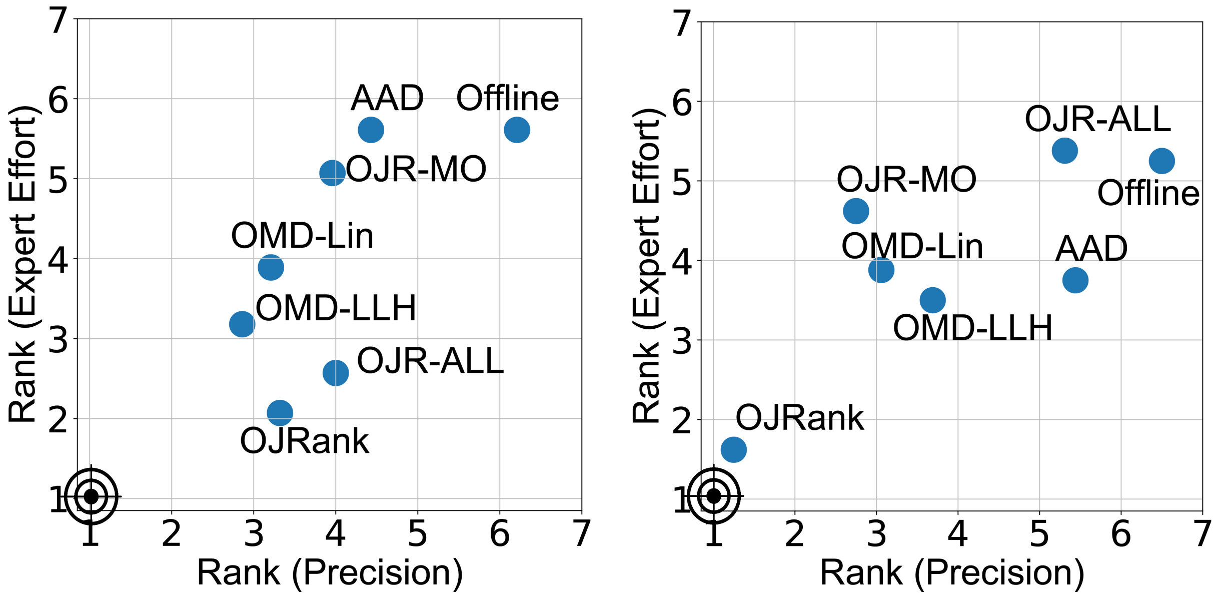
Figure 12.1: Illustration of OJRANK. Filled instances are true anomalies, unfilled are nominals, color depicts similarity. Upon each feedback, OJRANK re-ranks the instances, aiming to (a) push up similar True-Positives (red filled) & (b) 'mute' similar False-Positives (orange unfilled); (a) helps reduce expert effort, and both (a,b) increase true positive rate.

novelties in particle physics experiments, it is also necessary for the human expert to validate the anomalies before claiming a discovery. In these types of anomaly mining settings, the human expert essentially produces a label (true or false positive) through verification of each next, yet-unverified top-1 instance.

In this work, our aim is to leverage each label feedback to update our detection model to produce a new ranking. The idea is to interleave each feedback provided by the expert with a re-ranking by the updated model, which potentially also changes the top-1 instance the expert sees next. As such, this is a setting in which the detection model works alongside the expert, and learns to (re)rank *on the job* or in other words learns while the expert is working.

The goals of the re-ranking are two-fold. First is to improve the detection performance within the expert's verification budget. For example, if an auditor has the capacity and time to validate b invoices in a given day, the goal is to reach as high precision at b as possible. Intuitively, the more the number of detected anomalies, the higher is the return (i.e., savings from error and fraud) on investment (i.e., a day's of expert's work). Second goal is to reduce the expert's verification effort, which we define in terms of the similarity between consecutive instances expert gets to verify. Intuitively, the more similar instances they see in sequence, the lower would be the context switch and hence their verification effort. Besides those goals, the requirement is to update the ranking fast so as not to stall the expert waiting to be presented with the next top-1 instance. To these ends, we propose an On-the-Job (online) re-RANKing technique called OJRANK that employs a 'more-like-this' strategy upon a true positive feedback and 'less-like-this' strategy on encountering a false positive feedback, as illustrated in Figure 12.1 (see caption).

There exist related work on learning to rank from top-1 feedback for information retrieval tasks [45, 46]. However, due to the applications being different, their goals differ. Specifically,



(a) Average rank of seven comparison methods over two sets of data; (left) BENCHMARK and (right) CLUSTERED; with respect to two metrics of interest; *precision@b* and *expert effort*.

Metric	Baselines / Datasets	AAD [55]	OMD Lin[260]	OMD LLH[260]	OJR MO	OJR ALL
prec. @b	BENCHMARK	0.015	0.5	0.5	0.005	0.008
	CLUSTERED	0.003	0.007	0.027	0.003	0.003
expert effort	BENCHMARK	0.001	0.010	0.024	$1e-4$	0.014
	CLUSTERED	0.027	0.007	0.012	0.004	0.004

(b) p-values for Wilcoxon signed ranked test between OJRANK and baseline methods for *precision@b* and *expert effort* over two sets of data. Note that except two cases (shaded in gray), performance gains are significant at 0.05.

Figure 12.2: **OJRANK** outperforms simple as well as state-of-the-art baselines *significantly* for two metrics - *precision@b* and *expert effort*. (See §12.4 for details)

these work aim to learn from all the feedback to improve the performance of their *final* model. As such, they focus on metrics on the quality of the post-feedback ranked list whereas we aim to maximize precision on the instances labeled/verified by the expert. The two most related state-of-the-art work on feedback-based anomaly ranking are AAD [55, 56] and OMD [260]. They have used various loss functions and optimization algorithms for re-ranking anomalies based on top-1 feedback, toward improving precision at the budget, but *without any emphasis on expert effort*. OJRANK employs the same underlying tree-based ensemble detector as in these work and outperforms both AAD and OMD in terms of both precision and (especially) expert effort, as shown in Figure 12.2. We provide more details about the datasets and baselines in Section 12.4. To summarize, the contributions of this paper can be outlined as follows.

- **On-the-Job Learning to Re-rank Anomalies:** We address the problem of learning to re-rank anomalies on the job, i.e. while the expert is working toward verifying the top-ranked anomalies. Each verification of the top-1 instance produces a label, which our proposed OJRANK uses to update the ranking presented to the expert next. To this end, we employ a pairwise learning to rank objective coupled with a carefully-designed online gradient descent learning, where the update equation has a clear interpretation for the detection task.
- **Higher Precision, Lower Effort:** We demonstrate that OJRANK employs a 'more-like-this' update strategy upon receiving a true positive (TP) feedback, and a 'less-like-this' strategy upon a false positive (FP) feedback. Both help achieve higher detection precision as they respectively boost TPs and mute FPs. At the same time, 'more-like-this' updates enable *similar* anomalies to be pushed up in the rank order and shown consecutively, which helps reduce expert effort.
- **Time and Space Efficiency:** OJRANK updates ranking after every label feedback, during which the user stalls to be presented with the next top-1 instance to verify. We show that OJRANK's online updates take constant-time in complexity and are near-instantaneous empirically, where the re-ranking is done within one fifth of a second on average. Moreover, OJRANK requires only linear space on the number of instances.

The code, datasets used in the experiments and supplementary information is available at https://ojrank.github.io.

12.1 Related Work

We discuss related work in two categories: *active sampling* (which carefully selects the instances to be labeled) versus *top-1 feedback* (which simply selects the top instance—no strategy is involved). We note that active learning (AL) and rare category discovery (RCD) fall under the former category, while on-the-job learning (OJL) is different and falls under the latter. Table 12.1 shows a quick comparison between OJRANK and various active sampling and top-1 feedback methods. Detailed discussion follows.

Active Sampling: AL is the task of selecting a small budget of most informative instances to be queried for labels, such that a model trained on those labeled instances achieves high performance. AL for classification has employed various selection/sampling strategies such as uncertainty, query-by-committee, variance reduction, etc. for the details of which we refer to a

Table 12.1: Qualitative comparison between OJRank and related methods.

Properties	Top-1 Feedback	Online Model Updates	Precision @ Budget	Expert Effort
AL [61, 90, 121, 182, 220, 261]	✗	✗	✗	✗
RCD [114, 115, 226]	✗	✗	✗	✗
Ghani and Kumar [80]	✗	✗	✓	✓
Top-1L2R [45, 46]	✓	✓	✗	✗
AAD [55, 56]	✓	✗	✓	✗
OMD [260]	✓	✓	✓	✗
OJRank	✓	✓	✓	✓

survey by Settles [251]. Others studied active sampling for learning-to-rank [61, 121, 182, 261] and for anomaly detection [90, 220, 226]. A key difference is in how the queries are selected: In active sampling they are strategically and carefully chosen; in OJL the query is always the top-most yet-unlabeled instance (i.e., there is no active selection). The goals also differ: active sampling aims to maximize the label-incorporated model's final performance on unlabeled examples, i.e. performance is measured *after* querying; in contrast OJL aims to maximize the number of anomalous instances presented to user *during* querying.

Active sampling techniques have also been studied for rare category discovery (RCD) [114, 115, 226], which is a setting where anomalies are assumed to form multiple micro-clusters (i.e. rare categories). The goal is also notably different from OJL's, where they aim to identify at least one example from each rare category by (strategically) querying the expert for as few labels as possible in total.

Ghani and Kumar [80] studied interactively detecting errors in insurance claims, while also aiming to reduce context switching costs for experts. They use a query selection heuristic that first clusters the top-scoring instances based on similarity and ranks the clusters based on a combination of measures. Instances from top-ranked cluster is then shown to the expert (helping reduce context switch) until precision falls below a threshold upon which instances from the next cluster are presented. Their model is never updated. In contrast, OJRank boosts up instances similar to a true-positive feedback in an online fashion.

Top-1 Feedback: Compared to active sampling, there has been relatively limited work on learning-to-rank (L2R) problems where feedback is only given for the topmost instance of the ranked list. Chaudhuri et al. [45, 46] proposed algorithms for well-known L2R loss functions from the pointwise, pairwise and listwise families. However, they aim to maximize the *resulting* performance over the entire ranked list after all feedback is collected, unlike in our setting, where our goal is to maximize the overall number of anomalies presented to the expert *during* feedback.

On the anomaly ranking side, Das et al. [55] proposed AAD (Active[1] Anomaly Discovery) that partly share the same goal as OJRank, that is to maximize the number of anomalies presented at the top (and partly not, as we also care about expert effort). After each feedback, AAD solves an optimization program involving constraints between *all* of the previous expert-labeled anomalies and nominals (i.e., updates are not online). As the number of pairs grows along with

[1] Here, their use of the term 'active' is misleading, as AAD does not employ any active sampling strategy for querying—always the top-1 instance is labeled. We find OJL to be a more suitable name, as the expert verifies the next top-1 instance sequentially as part of their job.

each feedback round, the all-pairs constraints increase the running time. The initial AAD method was intended to work with LODA [228], a projection-based ensemble detection algorithm, which is later extended [56] for tree-based ensembles like iForest [179] and HS-Trees [275]. AAD is also sped up by intelligently replacing the all-pairs constraints by those relative to the instance ranked at the τ^{th}-quantile.

Most recently, Siddiqui et al. [260] proposed to optimize pointwise loss functions in an online fashion upon each feedback via online mirror descent (OMD). AAD and OMD (and variants) all aim to maximize the number of true anomalies shown to the expert. However, they do not put any emphasis on expert effort, which takes into account the effort consumed while verifying instances that is likely to decrease if the instances shown consecutively are similar.

12.2 Preliminaries and Problem Definition

12.2.1 Learning on-the-job Setup

We are given a dataset containing n instances $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in d dimensions, as well as an anomaly detection model M that provides scores for the input instances $\{s_1, \ldots, s_n\}$, the higher the more anomalous. The instances are ranked in descending order of these scores.

The procedure of learning on-the-job proceeds in rounds. In each round, (a) the expert verifies the top-1 instance with the highest score and reveals a label and (b) our OJRANK algorithm uses this feedback to update the detection model M and hence the ranking of the instances. In the next round, the expert is presented with the top-1 instance based on the *updated* ranking and so on. This procedure continues for b rounds, where b specifies the expert's budget (e.g., the number of invoices an auditor has the capacity to analyze within a day's work).

12.2.2 Family of Detection Models

Our proposed work can subsume any *ensemble* anomaly detection model M, where each ensemble component provides a separate score and the overall anomalousness score of an instance is the sum (or average) of those scores across components. Many state-of-the-art detectors fall into this category such as LOF with feature bagging [166], LODA [228] with 1-d projections, and various tree-based ensembles including iForest [180], HS-Trees [275], and RS-Forest [301].

Without loss of this generality, this paper adopts the iForest detector. We denote the number of components (i.e., iTrees) in the ensemble by m. Each iTree is constructed over a random subsample of the input data $\mathcal{D}$, by splitting the data at each internal node over a randomly selected feature and a threshold. As anomalies are fewer in number and isolated from nominal instances, they require fewer splits to reach a leaf node, hence are quickly isolated. Therefore, anomalous instances are located at a shorter depth than nominal instances on average over all the trees in the iForest.

We denote the number of leaves in tree t by L_t. Each instance is placed in exactly one of the L_t leaves in each tree. The score of an instance u by the t-th tree is given by

$$s_u^{(t)} = 1/[\ \text{pathlen}(l_u^{(t)}) + h(cnt_{l_u^{(t)}})\ ] \tag{12.1}$$

where $l_u^{(t)}$ denotes the leaf in t which u falls into; pathlen captures its depth from the root, $cnt_{l_u^{(t)}}$ is the total number of instances it contains, and h function returns the expected path length of unsuccessful searches in a Binary Search Tree (BST) constructed with the given number of samples.

In this paper we work with the leaves representation denoted by $\mathbf{s}_u = [\mathbf{s}_u^{(1)} \dots \mathbf{s}_u^{(m)}]$ where $\mathbf{s}_u^{(t)}$ is a vector with entries $i = 1 \dots L_t$ where

$$\mathbf{s}_u[i] = \begin{cases} s_u^{(t)}, & \text{if } i = l_u^{(t)} \\ 0, & \text{otherwise} \end{cases} \tag{12.2}$$

As such $\mathbf{s}_u$ is $\sum_{t=1}^{m} L_t = l$ dimensional with exactly m nonzeros. We denote by $\boldsymbol{S} \in \mathbb{R}^{n \times l}$ the scores matrix. As such, $\mathbf{s} = \boldsymbol{S} \cdot \mathbf{1}$ contains all the anomaly scores.

12.2.3 Metrics of Interest and Problem Statement

In this work, we aim to improve two different metrics of interest.

First is the total number of true anomalies verified by the expert within their budget b. In auditing systems, this would correspond to the number of erroneous (tax, insurance, reimbursement) invoices caught among the ones they could analyze within a day's work—others that could not be analyzed need to be paid in full—as such, the more errors caught, the higher the savings could be. This metric is essentially the precision at the budget, denoted *precision@b*.

The second metric of interest is related to the cognitive burden the expert would have due to context switch. Intuitively, the more similar two instances analyzed in sequence are, the lower the context switching costs would be for the expert. Since expert effort is not as well-established a metric as precision, we define it as follows.

Definition 9 (***expert effort***)**.** *: Given the sequence of b instances $\{\mathbf{s}_{\pi(1)}, \dots, \mathbf{s}_{\pi(b)}\}$, where $\pi(r)$ denotes the index of the instance ranked at the top in round r, we define:*

$$\textit{expert effort} = \sum_{r=1}^{b-1} 1 - \textit{sim}\big(\mathbf{s}_{\pi(r)}, \mathbf{s}_{\pi(r+1)}\big) \ , \tag{12.3}$$

which is the similarity between consecutive instances verified by the expert. Here we employ cosine similarity in the scoring space $\mathcal{S}$. If two instances fall in the same leaves with high scores (see Eq. (12.1)), these points are considered anomalous for the same reasons (in the same feature subspaces). Intuitively, analyzing invoices containing similar type of anomalies would reduce verification effort.

One can also argue for similarity in the input space $\mathcal{X}$. This captures the insight that two similar-looking invoices would be easier for the expert to process back to back. In the experiments we show that OJRANK outperforms the baselines w.r.t. both similarities.

Having outlined the preliminaries and goals, our problem statement can be given as follows:

Problem 1 (**On-the-Job Re-ranking**)**.** *For rounds $1 \dots b$:*

- ***Obtain*** *label for the top-1 instance from expert*
- ***Update*** *the detection model based on the feedback and re-rank instances*

such that precision@b is maximized, total expert effort is minimized, and updates are fast.

We set up the model update problem as learning a ranking of the instances based on a *weighted* sum of leaf scores. That is, we replace the sum $\mathrm{s} = \boldsymbol{S} \cdot \mathbf{1}$ with

$$\mathrm{s} = \boldsymbol{S} \cdot \mathbf{w} \tag{12.4}$$

and aim to estimate $\mathbf{w}$ from expert feedback on-the-job.[2]

12.3 Proposed Approach: OJRank

We formulate the on-the-job re-ranking problem as an online learning-to-rank task. To this end, we adopt a pairwise learning to rank objective with a convex cross entropy loss.

Given training examples $\langle (u, v), p_{uv} \rangle \in T$ where p_{uv} is the desired probability of instance u being ranked above instance v, we aim to find the weight vector that minimizes the cross entropy loss over all the training pairs:

$$\min_{\mathbf{w}} f = \sum_{(u,v) \in T} -p_{uv} \log(\widehat{p}_{uv}) - (1 - p_{uv}) \log(1 - \widehat{p}_{uv}) \tag{12.5}$$

where $\widehat{p}_{uv}$ is the estimated probability based on our current estimate of $\mathbf{w}$, and is acquired using the logistic function:

$$\widehat{p}_{uv} = \frac{e^{(s_u - s_v)}}{1 + e^{(s_u - s_v)}}\ , \text{ where } s_u = \mathrm{s}[u] = \boldsymbol{S}_u \cdot \mathbf{w} \tag{12.6}$$

Updating $\mathbf{w}$ leads to updating estimated probability $\widehat{p}_{uv}$ and moving it closer to the desired probability p_{uv}.

OJRank re-ranks the anomalies after obtaining the feedback on top-1 instance u from the expert. It has two major components: (1) **Generating pairs** - as the cross entropy loss function is pairwise, we pair u for which we received feedback with other instances, which we choose by either sampling or using the historical instances labeled in the previous rounds, and (2) **Optimization** - which involves updating $\mathbf{w}$ via optimizing the loss function over the generated pairs, which is then used to re-compute the scores s in the following round. The top-1 instance from the updated ranking is presented to the expert for feedback. The steps of the algorithm are given in Algorithm 4.

12.3.1 Generating pairs

After each round of feedback, we obtain label from the expert for a single instance u - we pair this instance with other instances v to create pairs.

Using history: We pair u with each *previously labeled* instance v with an opposite label to that of u. This allows us to establish a clear ordering amongst paired instances - indicating which instance should be ranked higher. For example, if u is anomalous (nominal), we pair it with nominal (anomalous) instances v, hence we are certain that u should be ranked higher (lower) than v.

[2] Using leaves representation provides us with the capacity to weight l different feature subspaces (that each leaf corresponds to) rather than m different trees, the former allowing a larger granularity as $l > m$.

By sampling: During initial feedback rounds, it is possible that there are no instances in the history that have opposite label to that of u. To handle such cases, we sample v from unlabeled instances such that we maximize the probability of obtaining oppositely labeled instances. Specifically, we skew the probability of sampling from unlabeled instances such that chances of getting oppositely labeled instances increase. This is done by (i) truncating the sample space – if u is anomalous (nominal), we sample from bottom (top) half of the ranked list and (ii) sampling an instance with probability inversely (directly) proportional to its score. In particular, when u is anomalous we use the sampling probability proportional to $1/s_v$ for instance v. If u is nominal, we sample v (after normalizing the scores $\bar{s}_v \in [0, 1]$) with probability proportional to $(c\bar{s}_v + 1)^{1/c}$ with $c = -0.99$ to increase the chances of sampling an anomalous v. The polynomial scaling for the latter is to account for the fewer number of anomalies in the data.

We give priority to generating pairs using history rather than sampling, as sampling could lead to pairing identically labeled instances. Therefore, we place an upper limit k on the number of sampled pairs and only when the number of pairs generated from history is less than (small) k, we sample the remaining pairs (lines 9, 14 in Algo. 4).

Besides the pairs (u, v), we also need to provide desired probability p_{uv} for each pair as input to the objective function in (12.5). For v that has been sampled from history and u anomalous (nominal), we set p_{uv} to be the maximum (minimum) of all the current estimated probability values among all pairs (lines 8 and 13). For example, if u is anomalous (v is nominal), we set $p_{uv} = \widehat{p}_{az}$ where a is the highest scored instance and z is the lowest scored instance (line 4). Here, we are certain about the ordering among the (u, v) instances, thus we set the desired probability to maximum so as to push u in the correct direction quickly. On the other hand, when v is obtained by sampling, there is still a chance that we might have generated identically labeled instances. In that case, we aim to avoid the mistake of pushing u in opposite direction by a high magnitude. Therefore, for sampled pairs we nudge the current estimate of probability between u and v by only a (small) factor of δ, i.e., $p_{uv} = (1 \pm \delta)\widehat{p}_{uv}$, the sign depending on whether u is anomalous or nominal (lines 10, 15).

12.3.2 Optimization

We now have training instances in the form of $\langle (u, v), p_{uv} \rangle \in P = \widetilde{P}_H \cup P_S$, generated using history and via sampling as explained in the previous subsection. Next, we are interested in solving the optimization problem in (12.5), i.e., find $\mathbf{w}$ such that the cross entropy loss over all pairs in the training set is minimized. The gradient update equation is written as

$$\mathbf{w}^t = \mathbf{w}^{t-1} - \eta \cdot \sum_{(u,v) \in P} (\widehat{p}_{uv} - p_{uv})(\boldsymbol{S}_u - \boldsymbol{S}_v) \; . \tag{12.7}$$

Relation of gradient updates to precision and effort: Importantly, these gradient updates are interpretable and have clear impact on the expert effort and the true positive rate. Consider the case where u is anomalous and v is nominal: by construction, we know that $p_{uv} > \widehat{p}_{uv}$. Each coordinate in $\mathbf{w}$ represents the relative importance of a leaf (i.e., subspace) in a tree. Looking at the update, we can observe that all the coordinates that are responsible for making u anomalous, i.e., those with a magnitude in $\boldsymbol{S}_u$ significantly higher than those in $\boldsymbol{S}_v$, will increase in weight. Therefore, the updated $\mathbf{w}$ will push u, as well as other anomalous instances similar to u (i.e.,

Algorithm 4 Proposed OJRank

```
Require: Ensemble (in our implementation iForest) scores $\boldsymbol{S} \in \mathbb{R}^{n \times l}$; Initial weights $\mathbf{w} \in \mathbb{R}^{l}$; Budget $b$; Scale
    factor $\delta$; Num. pairs to sample $k$
 1: $\mathbf{s} = \boldsymbol{S} \cdot \mathbf{w}$; round $= 0$                                  ▷ Initialize
 2: while round $< b$ do
 3:     $y_u$ ←label from expert for $u := \arg\max(\mathbf{s})$                        ▷ Top-1 feedback
    /* Setting Up (Generating Pairs)*/
 4:     $a \leftarrow \arg\max(\mathbf{s}), z \leftarrow \arg\min(\mathbf{s})$
 5:     $P_H = \emptyset, P_S = \emptyset$                                     ▷ Historical and Sampled sets
 6:     if $y_u = 1$ then                                         ▷ True anomaly (positive)
 7:         for $v \in \mathcal{H}_N$ do
 8:             add $\langle (u,v), \widehat{p}_{az} \rangle$ to $P_H$
 9:         for $v \in$ sample$(\mathbf{s}, y_u, (k - |\mathcal{H}_N|)_+)$ do
10:             add $\langle (u,v), (1+\delta)\widehat{p}_{uv} \rangle$ to $P_S$
11:     else                                                       ▷ False positive
12:         for $v \in \mathcal{H}_A$ do
13:             add $\langle (u,v), (1 - \widehat{p}_{az}) \rangle$ to $P_H$
14:         for $v \in$ sample$(\mathbf{s}, y_u, (k - |\mathcal{H}_A|)_+$ do
15:             add $\langle (u,v), (1-\delta)\widehat{p}_{uv} \rangle$ to $P_S$
    /* Optimization (Updating weights)*/
16:     $t = 0$;  $\mathbf{w}^t = \mathbf{w}^{t-1} \leftarrow \mathbf{w}$                     ▷ Initialize with $\mathbf{w}$ from last round
17:     $\eta = 0.1, \gamma = 0.75, \epsilon = 10^{-8}$, batch_size $= 100, T_{\max} = 1000$
18:     repeat
19:         $\widetilde{P}_H \leftarrow$ get_next_SGD_batch$(P_H,$ batch_size$)$
20:         for $\langle (u,v), p_{uv} \rangle \in \widetilde{P}_H \cup P_S$ do
21:             if $(u,v) \in P_S$ then $c \leftarrow \mathbb{1}(\boldsymbol{S}_u > 0)$ else $c \leftarrow [1 \ldots l]$
22:             $\mathbf{w}^{t+1}[c] \leftarrow \mathbf{w}^t[c] - \gamma(\mathbf{w}^t[c] - \mathbf{w}^{t-1}[c])$
                            $- \eta(\boldsymbol{S}_u[c] - \boldsymbol{S}_v[c])(\widehat{p}_{uv} - p_{uv})$
23:         $t \leftarrow t + 1$
24:     until $t \geq T_{\max}$ OR $f(\mathbf{w}^{t+1}) - f(\mathbf{w}^t) \leq \epsilon$
25:     $\mathbf{w} := \mathbf{w}^{t+1}$,  $\mathbf{s} = \boldsymbol{S} \cdot \mathbf{w}$                      ▷ Rescore according to the updated $\mathbf{w}$
26:     if $y_u == 1$ then $\mathcal{H}_A := \mathcal{H}_A \cup u$ else $\mathcal{H}_N := \mathcal{H}_N \cup u$
27:     round ← round + 1
```

those that share the same high-scoring leaves with u) higher to the top; contributing to reduced effort and increased precision. In contrast, $p_{uv} < \widehat{p}_{uv}$ when u is nominal, in which case updates will tend to push u and other similar instances down in ranking, contributing to reduced false positive rate.

Coordinate selection: One caveat with the gradient updates is the set of *sampled* pairs, which may contain identically-labeled (u, v) pairs. In those cases, the updates will nevertheless enforce a (wrong) ordering between them. For example, if u is nominal and we sampled v nominal as well, updates will tend to increase weights on coordinates of v that have higher magnitude than those of u (e.g., different leaf in the same tree), causing v (and nominals similar to v) climb higher in the list, which is undesirable. We circumvent this issue by updating only the non-zero coordinates of u for sampled pairs (as shown in line 21).

Online updates and acceleration: Note that $\mathbf{w}^0$ is set to the latest $\mathbf{w}$ from the previous round (lines 16, 25), and the updates are only over the newly generated pairs P for the top-1 labeled instance u in the current round (line 20). As such, OJRank stands on *online* gradient-based learning.

Specifically, we employ batch stochastic gradient descent (SGD). Each batch is selected from the pairs that are created using history (line 19) and combined with the (at most k) sampled pairs. We also use the momentum-based SGD to accelerate the descent (hence, the response time). The momentum-based update equation (line 22) is similar to Eq. (12.7), which is obtained by adding γ fraction (momentum factor) of the gradient from the previous step to the current gradient update vector. Finally, all relevant parameters are listed in line 17 of Algo 4 and our implementation is open-sourced at `https://ojrank.github.io`.

12.4 Evaluation

Table 12.2: Summary statistics for two sets of data used in experiments: (left) BENCHMARK and (right) CLUSTERED.

BENCHMARK DATASETS			CLUSTERED DATASETS		
Name	n	**Anom. %**	**Name**	n	**Anom. %**
abalone	1920	1.51	vowels	2821	1.77
ann	3251	2.24	optdigits	592	4.22
cardio	1700	2.64	letters	2433	2.05
ecoli	336	2.67	sensor	16257	0.92
glass	214	4.20	segment	1090	4.58
mammography	11183	2.32	statlog	1665	3.00
shuttle	12345	7.02	vehicle	495	6.06
wbc	378	5.56	svmguide	544	9.19
yeast	1191	4.61			
lympho	148	4.05			
musk	3062	3.16			
thyroid	3772	2.46			
wine	129	7.76			
vertebral	240	12.5			

In this section, we evaluate our proposed OJRANK approach in comparison with five baselines as listed in §12.4.1. Experiments are conducted over two types of datasets, introduced in §12.4.2. We then present the results with respect to three performance metrics of interest in §12.4.3: *precision@b*, *expert effort* and speed (i.e., online response time).

12.4.1 Baselines

We compare OJRANK with two state-of-the-art techniques that addressed the problem of online re-ranking of anomalies from top-1 feedback, as discussed in related work (§12.1). We also compare with the offline baseline as well as two variants of OJRANK explained below.

- **AAD [55]**: See §12.1 and Table 12.1.

- **OMD [260]**: See §12.1 and Table 12.1. We compare to both versions based on the type of loss used – (a) OMD-Lin (linear loss) and (b) OMD-LLH (log-likelihood loss).
- **Offline**: Static top-b instances based on the initial ranking by the detector, no re-ranking over rounds.
- **OJR-MO**: Mistake-Only variant; we run online model updates only when top-1 feedback is a false positive.
- **OJR-ALL**: All coordinates variant; we do not scale the sampling probabilities–increases the risk of identically-labeled pairs (`ilp`)–and perform no coordinate selection for sampled pairs–enforces a ranking among `ilp`.

All compared methods use the same underlying iForest detectors. We report performance results averaged over 10 different runs of iForest. AAD and OMD are run with the author-recommended parameters. We set budget b equal to the number of true anomalies in each dataset.

Table 12.3: ***precision@b* on BENCHMARK DATASETS.** Per dataset rank provided in parentheses (the lower the better). Average rank across datasets given in the last row. Symbols ▲ and △ denote the cases where OJRANK is significantly better than the baseline w.r.t. the Wilcoxon signed rank test, respectively at (p<0.01) and (p<0.05).

Dataset	OJRANK	**OJR-MO**	**OJR-ALL**	**AAD**	**OMD-Lin**	**OMD-LLH**	**Offline**
abalone	0.52 ± 0.00(5.0)	0.52 ± 0.00(5.0)	0.52 ± 0.00(5.0)	0.56 ± 0.02(1.0)	0.52 ± 0.01(3.0)	0.54 ± 0.02(2.0)	0.51 ± 0.02(7.0)
ann	0.75 ± 0.03(3.0)	0.73 ± 0.03(4.0)	0.52 ± 0.32(5.0)	0.39 ± 0.05(6.0)	0.78 ± 0.03(1.0)	0.76 ± 0.04(2.0)	0.19 ± 0.07(7.0)
cardio	0.64 ± 0.02(4.0)	0.65 ± 0.02(2.0)	0.60 ± 0.06(5.0)	0.55 ± 0.04(6.0)	0.65 ± 0.01(3.0)	0.69 ± 0.04(1.0)	0.38 ± 0.04(7.0)
ecoli	0.72 ± 0.06(1.0)	0.57 ± 0.08(3.0)	0.70 ± 0.10(2.0)	0.44 ± 0.05(6.0)	0.56 ± 0.09(4.0)	0.51 ± 0.10(5.0)	0.42 ± 0.04(7.0)
glass	0.11 ± 0.00(4.5)	0.11 ± 0.00(4.5)	0.17 ± 0.17(1.0)	0.11 ± 0.00(4.5)	0.11 ± 0.00(4.5)	0.11 ± 0.00(4.5)	0.11 ± 0.00(4.5)
mammography	0.58 ± 0.02(3.0)	0.56 ± 0.02(5.0)	0.56 ± 0.01(4.0)	0.41 ± 0.02(6.0)	0.60 ± 0.01(2.0)	0.62 ± 0.01(1.0)	0.25 ± 0.05(7.0)
shuttle	0.96 ± 0.04(5.0)	0.94 ± 0.03(6.0)	0.97 ± 0.01(4.0)	0.98 ± 0.00(1.0)	0.97 ± 0.01(3.0)	0.98 ± 0.00(2.0)	0.89 ± 0.03(7.0)
wbc	0.71 ± 0.06(1.0)	0.66 ± 0.03(3.0)	0.67 ± 0.05(2.0)	0.53 ± 0.05(6.0)	0.60 ± 0.03(4.0)	0.59 ± 0.05(5.0)	0.50 ± 0.03(7.0)
yeast	0.27 ± 0.05(5.0)	0.25 ± 0.05(6.0)	0.18 ± 0.04(7.0)	0.34 ± 0.02(3.0)	0.35 ± 0.01(2.0)	0.36 ± 0.04(1.0)	0.34 ± 0.01(4.0)
lympho	0.92 ± 0.08(6.0)	0.93 ± 0.08(3.0)	0.60 ± 0.11(7.0)	0.93 ± 0.08(3.0)	0.93 ± 0.08(3.0)	0.93 ± 0.08(3.0)	0.93 ± 0.08(3.0)
musk	1.00 ± 0.00(3.0)	0.99 ± 0.00(4.0)	0.99 ± 0.01(6.0)	0.99 ± 0.02(5.0)	1.00 ± 0.00(1.5)	1.00 ± 0.00(1.5)	0.97 ± 0.03(7.0)
thyroid	0.81 ± 0.02(4.0)	0.77 ± 0.01(5.0)	0.82 ± 0.02(2.0)	0.69 ± 0.03(6.0)	0.82 ± 0.02(3.0)	0.86 ± 0.01(1.0)	0.54 ± 0.03(7.0)
wine	0.42 ± 0.19(1.0)	0.27 ± 0.13(3.0)	0.28 ± 0.35(2.0)	0.09 ± 0.03(5.5)	0.09 ± 0.03(5.5)	0.09 ± 0.03(5.5)	0.09 ± 0.03(5.5)
vertebral	0.33 ± 0.04(1.0)	0.31 ± 0.06(2.0)	0.05 ± 0.05(4.0)	0.05 ± 0.02(3.0)	0.05 ± 0.02(5.5)	0.05 ± 0.02(5.5)	0.04 ± 0.02(7.0)
Avg. Rank	**3.32**	**3.96**$^{\blacktriangle}$	**4.00**$^{\blacktriangle}$	**4.43**$^{\triangle}$	**3.21**	**2.86**	**6.21**$^{\blacktriangle}$

Table 12.4: ***precision@b* on CLUSTERED DATASETS.** Per dataset rank provided in parentheses (lower is better). Average rank provided in the last row. Symbol ▲ denote the cases where OJRANK is significantly better than the corresponding baseline w.r.t. the Wilcoxon signed rank test at (p<0.01).

Dataset	OJRANK	**OJR-MO**	**OJR-ALL**	**AAD**	**OMD-Lin**	**OMD-LLH**	**Offline**
vowels	0.78 ± 0.09(2.0)	0.58 ± 0.09(4.0)	0.09 ± 0.26(7.0)	0.45 ± 0.04(5.0)	0.77 ± 0.06(3.0)	0.80 ± 0.05(1.0)	0.17 ± 0.06(6.0)
optdigits	0.08 ± 0.13(1.0)	0.07 ± 0.13(2.0)	0.01 ± 0.03(7.0)	0.04 ± 0.03(5.0)	0.06 ± 0.04(3.0)	0.05 ± 0.03(4.0)	0.03 ± 0.02(6.0)
letters	0.62 ± 0.12(1.0)	0.51 ± 0.12(3.0)	0.21 ± 0.28(5.0)	0.16 ± 0.06(6.0)	0.53 ± 0.12(2.0)	0.47 ± 0.17(4.0)	0.05 ± 0.01(7.0)
sensor	0.95 ± 0.04(1.0)	0.95 ± 0.03(2.0)	0.48 ± 0.38(6.0)	0.52 ± 0.12(5.0)	0.95 ± 0.03(4.0)	0.95 ± 0.03(3.0)	0.14 ± 0.08(7.0)
segment	0.48 ± 0.20(1.0)	0.40 ± 0.14(2.0)	0.00 ± 0.00(6.5)	0.02 ± 0.02(5.0)	0.25 ± 0.15(3.0)	0.04 ± 0.03(4.0)	0.00 ± 0.00(6.5)
statlog	0.93 ± 0.02(2.0)	0.91 ± 0.01(5.0)	0.92 ± 0.01(4.0)	0.90 ± 0.01(6.0)	0.93 ± 0.01(1.0)	0.92 ± 0.01(3.0)	0.87 ± 0.03(7.0)
vehicle	0.31 ± 0.14(1.0)	0.29 ± 0.07(2.0)	0.12 ± 0.03(4.0)	0.11 ± 0.03(6.0)	0.13 ± 0.04(3.0)	0.11 ± 0.03(5.0)	0.09 ± 0.02(7.0)
svmguide	0.12 ± 0.04(1.0)	0.11 ± 0.01(2.0)	0.10 ± 0.03(3.0)	0.10 ± 0.00(5.5)	0.10 ± 0.00(5.5)	0.10 ± 0.00(5.5)	0.10 ± 0.00(5.5)
Avg. Rank	**1.25**	**2.75**$^{\blacktriangle}$	**5.31**$^{\blacktriangle}$	**5.44**$^{\blacktriangle}$	**3.06**$^{\blacktriangle}$	**3.69**$^{\blacktriangle}$	**6.50**$^{\blacktriangle}$

12.4.2 Datasets

We evaluate performance over two types of datasets as listed in Table 12.2; namely, (1) BENCHMARK DATASETS: a set of 14 real-world datasets and (2) CLUSTERED DATASETS: 8 datasets generated from multi-class classification datasets, as described below.

BENCHMARK DATASETS: The first data collection contains 14 real-world datasets from a publicly-available outlier detection dataset repository [237].

CLUSTERED DATASETS: OJRank learns well from feedback when there are other points similar to the feedback instance, i.e., when instances form (micro)-clusters. Learning from an extreme outlier would be very limited, as there are no other points in the dataset that are similar to it, hence no other instances can benefit from the feedback. To create such a setting, we synthetically generate 8 datasets by modifying multi-class datasets from the UCI repository.

From each multi-class dataset, we first select two classes at random, with the intuition that the instances within each class would be clustered. Instances from the remaining classes are designated as nominals. In case of too many remaining classes, 3 of them are randomly selected. We next downsample the selected two classes to equal number so that the percentage is consistent with the usual anomaly detection settings. We designate the downsampled instances from the first class as "anomalies" and those from the other as "rare nominals". The detector is likely to rank both as anomalous, yet from the expert's point of view, they would correspond to true and false positives, respectively. Here, "rare nominals" represent rare yet uninteresting group of instances. This setup allows us to directly test the ability of the methods in learning to boost/mute instances from these respective classes upon expert feedback.

Table 12.5: List of CLUSTERED DATASETS. We list the type of instances and from what class were they sampled.

Dataset	# Instances	Anom. %	Description
Vowels	2821	1.77	Anomaly (Class 4[25]) Rare Nominals (Class 8[25]) Frequent Nominals (Class 2,3,6)
Optdigits	592	4.22	Anomaly (Class 8[25]) Rare Nominals (Class 2[25]) Frequent Nominals (Class 1,3,5)
Letters	2433	2.05	Anomaly (Class 25[50]) Rare Nominals (Class 7[50]) Frequent Nominals (Class 20, 3, 15)
Sensor	16257	0.92	Anomaly (Class 8[150]) Rare Nominals (Class 5[150]) Frequent Nominals (Class 1, 7, 9)
Segment	1090	4.58	Anomaly (Class 1[50]) Rare Nominals (Class 2[50]) Frequent Nominals (Class 5, 6, 7)
Statlog	1665	3.00	Anomaly (Class 2[50]) Rare Nominals (Class 4[50]) Frequent Nominals (Class 1, 3, 5, 7)
Vehicle	495	6.06	Anomaly (Class 0[30]) Rare Nominals (Class 3[30]) Frequent Nominals (Class 1, 2)
Svmguide	544	9.19	Anomaly (Class +3[50]) Rare Nominals (Class -3[50]) Frequent Nominals (Class -2, -1, 1, 2)

Summary statistics for the CLUSTERED DATASETS are given in Table 12.2. Details for the mapping of classes to above categories are provided in Table 12.5. We also share these generated datasets at our aforementioned URL.

Table 12.6: ***Expert effort* on BENCHMARK DATASETS.** Per dataset rank shown in parentheses (lower is better). Average rank is in the second last row (*effort* in $\mathcal{S}$ space). Average rank for *effort* in $\mathcal{X}$ space also given in last row. Symbols $\blacktriangle$(p<0.01) and $\triangle$(p<0.05) denote the cases where OJRank is significantly better than the baseline w.r.t. Wilcoxon signed rank test.

Dataset	OJRank	**OJR-MO**	**OJR-ALL**	**AAD**	**OMD-Lin**	**OMD-LLH**	**Offline**
abalone	0.59 ± 0.02(3.0)	0.65 ± 0.02(7.0)	0.60 ± 0.03(5.0)	0.60 ± 0.08(4.0)	0.54 ± 0.04(1.0)	0.56 ± 0.03(2.0)	0.65 ± 0.03(6.0)
ann	0.49 ± 0.02(1.0)	0.70 ± 0.02(5.0)	0.65 ± 0.21(4.0)	0.92 ± 0.02(7.0)	0.60 ± 0.02(3.0)	0.60 ± 0.02(2.0)	0.92 ± 0.02(6.0)
cardio	0.69 ± 0.01(1.0)	0.83 ± 0.03(5.0)	0.72 ± 0.03(2.0)	0.89 ± 0.03(6.0)	0.77 ± 0.02(4.0)	0.73 ± 0.03(3.0)	0.90 ± 0.03(7.0)
ecoli	0.96 ± 0.02(1.0)	1.10 ± 0.02(7.0)	0.98 ± 0.02(2.0)	1.09 ± 0.02(5.0)	1.07 ± 0.02(4.0)	1.07 ± 0.04(3.0)	1.10 ± 0.02(6.0)
glass	1.12 ± 0.00(7.0)	1.12 ± 0.00(6.0)	1.09 ± 0.07(1.0)	1.12 ± 0.00(3.5)	1.12 ± 0.00(3.5)	1.12 ± 0.00(3.5)	1.12 ± 0.00(3.5)
mammography	0.57 ± 0.01(2.0)	0.70 ± 0.01(5.0)	0.60 ± 0.01(3.0)	0.82 ± 0.02(6.0)	0.60 ± 0.02(4.0)	0.57 ± 0.02(1.0)	0.83 ± 0.02(7.0)
shuttle	0.20 ± 0.03(2.0)	0.65 ± 0.04(5.0)	0.20 ± 0.01(1.0)	0.81 ± 0.02(7.0)	0.35 ± 0.01(4.0)	0.26 ± 0.01(3.0)	0.70 ± 0.03(6.0)
wbc	0.91 ± 0.02(1.0)	1.00 ± 0.01(5.0)	0.92 ± 0.02(2.0)	1.00 ± 0.01(6.0)	0.99 ± 0.01(4.0)	0.98 ± 0.01(3.0)	1.00 ± 0.01(7.0)
yeast	0.86 ± 0.03(4.0)	0.91 ± 0.03(6.0)	0.91 ± 0.02(7.0)	0.81 ± 0.03(3.0)	0.77 ± 0.01(2.0)	0.77 ± 0.02(1.0)	0.88 ± 0.01(5.0)
lympho	1.20 ± 0.00(2.0)	1.20 ± 0.00(5.0)	1.19 ± 0.00(1.0)	1.20 ± 0.00(5.0)	1.20 ± 0.00(5.0)	1.20 ± 0.00(5.0)	1.20 ± 0.00(5.0)
musk	0.20 ± 0.01(1.0)	0.47 ± 0.03(5.0)	0.22 ± 0.01(2.0)	0.48 ± 0.03(7.0)	0.26 ± 0.02(4.0)	0.24 ± 0.01(3.0)	0.48 ± 0.02(6.0)
thyroid	0.48 ± 0.02(2.0)	0.66 ± 0.04(5.0)	0.48 ± 0.02(1.0)	0.80 ± 0.03(7.0)	0.59 ± 0.03(4.0)	0.51 ± 0.03(3.0)	0.76 ± 0.02(6.0)
wine	1.06 ± 0.04(1.0)	1.10 ± 0.01(3.0)	1.06 ± 0.05(2.0)	1.11 ± 0.00(7.0)	1.11 ± 0.00(5.0)	1.11 ± 0.00(6.0)	1.10 ± 0.00(4.0)
vertebral	0.89 ± 0.06(1.0)	0.95 ± 0.02(2.0)	1.00 ± 0.02(3.0)	1.02 ± 0.00(5.0)	1.02 ± 0.00(7.0)	1.02 ± 0.00(6.0)	1.02 ± 0.00(4.0)
Avg. Rank	**2.07**	**5.07**$^{\blacktriangle}$	**2.57**$^{\triangle}$	**5.61**$^{\blacktriangle}$	**3.89**$^{\triangle}$	**3.18**$^{\triangle}$	**5.61**$^{\blacktriangle}$
Avg. Rank (Orig. Space)	**2.00**	**4.75**$^{\blacktriangle}$	**3.07**$^{\triangle}$	**5.57**$^{\blacktriangle}$	**3.86**$^{\triangle}$	**3.11**$^{\triangle}$	**5.64**$^{\blacktriangle}$

Table 12.7: ***Expert effort* on CLUSTERED DATASETS.** Per dataset rank provided in parentheses (lower is better). Average rank is in the second last row. Average rank for *effort* in $\mathcal{X}$ space also given in last row. Symbols $\blacktriangle$(p<0.01), $\triangle$(p<0.05) and $\triangledown$(p<0.1) denote the cases where OJRank is significantly better than the baseline w.r.t. Wilcoxon signed rank test.

Dataset	OJRank	**OJR-MO**	**OJR-ALL**	**AAD**	**OMD-Lin**	**OMD-LLH**	**Offline**
vowels	0.63 ± 0.06(1.0)	0.85 ± 0.03(4.0)	0.97 ± 0.13(7.0)	0.90 ± 0.02(5.0)	0.69 ± 0.02(3.0)	0.67 ± 0.02(2.0)	0.94 ± 0.01(6.0)
optdigits	1.02 ± 0.03(2.0)	1.03 ± 0.01(7.0)	1.03 ± 0.01(4.0)	0.99 ± 0.02(1.0)	1.03 ± 0.00(6.0)	1.03 ± 0.00(5.0)	1.02 ± 0.01(3.0)
letters	0.72 ± 0.06(1.0)	0.89 ± 0.04(4.0)	0.91 ± 0.14(5.0)	0.92 ± 0.04(6.0)	0.84 ± 0.06(2.0)	0.86 ± 0.07(3.0)	0.98 ± 0.01(7.0)
sensor	0.45 ± 0.05(1.0)	0.65 ± 0.06(4.0)	0.72 ± 0.22(5.0)	0.92 ± 0.01(7.0)	0.55 ± 0.05(3.0)	0.52 ± 0.06(2.0)	0.88 ± 0.05(6.0)
segment	0.71 ± 0.14(1.0)	0.83 ± 0.07(3.0)	1.01 ± 0.01(7.0)	0.80 ± 0.03(2.0)	0.88 ± 0.07(4.0)	0.95 ± 0.02(6.0)	0.95 ± 0.03(5.0)
statlog	0.52 ± 0.01(1.0)	0.66 ± 0.03(5.0)	0.54 ± 0.02(2.0)	0.69 ± 0.03(6.0)	0.60 ± 0.02(4.0)	0.56 ± 0.02(3.0)	0.69 ± 0.03(7.0)
vehicle	0.88 ± 0.07(1.0)	0.95 ± 0.04(3.0)	0.99 ± 0.01(7.0)	0.94 ± 0.02(2.0)	0.97 ± 0.02(5.0)	0.97 ± 0.02(4.0)	0.98 ± 0.02(6.0)
svmguide	0.95 ± 0.04(5.0)	0.97 ± 0.01(7.0)	0.95 ± 0.02(6.0)	0.88 ± 0.01(1.0)	0.94 ± 0.01(4.0)	0.92 ± 0.01(3.0)	0.91 ± 0.02(2.0)
Avg. Rank	**1.62**	**4.62**$^{\blacktriangle}$	**5.38**$^{\blacktriangle}$	**3.75**$^{\triangle}$	**3.88**$^{\blacktriangle}$	**3.50**$^{\triangle}$	**5.25**$^{\blacktriangle}$
Avg. Rank (Orig. Space)	**2.25**	**4.12**$^{\blacktriangle}$	**4.38**$^{\triangledown}$	**4.00**$^{\triangledown}$	**4.12**$^{\triangledown}$	**3.88**$^{\triangledown}$	**5.25**$^{\triangle}$

12.4.3 Results

We analyze performance results over both datasets w.r.t. (a) *precision@b*, (b) *expert effort* and (c) runtime per update. We also present a sensitivity analysis of OJRank w.r.t. two input parameters in Algo. 4; δ and k.

***Precision@b*:** Table 12.3 and Table 12.4 provide precision across BENCHMARK DATASETS and CLUSTERED DATASETS, respectively. On each dataset, we show the average precision and standard deviation over 10 different runs of iForest. Rank of each method per dataset is in parentheses (in case of ties, average of the ranks are assigned to each tied method). Finally, the last row gives the average rank per method across all datasets (lower is better).

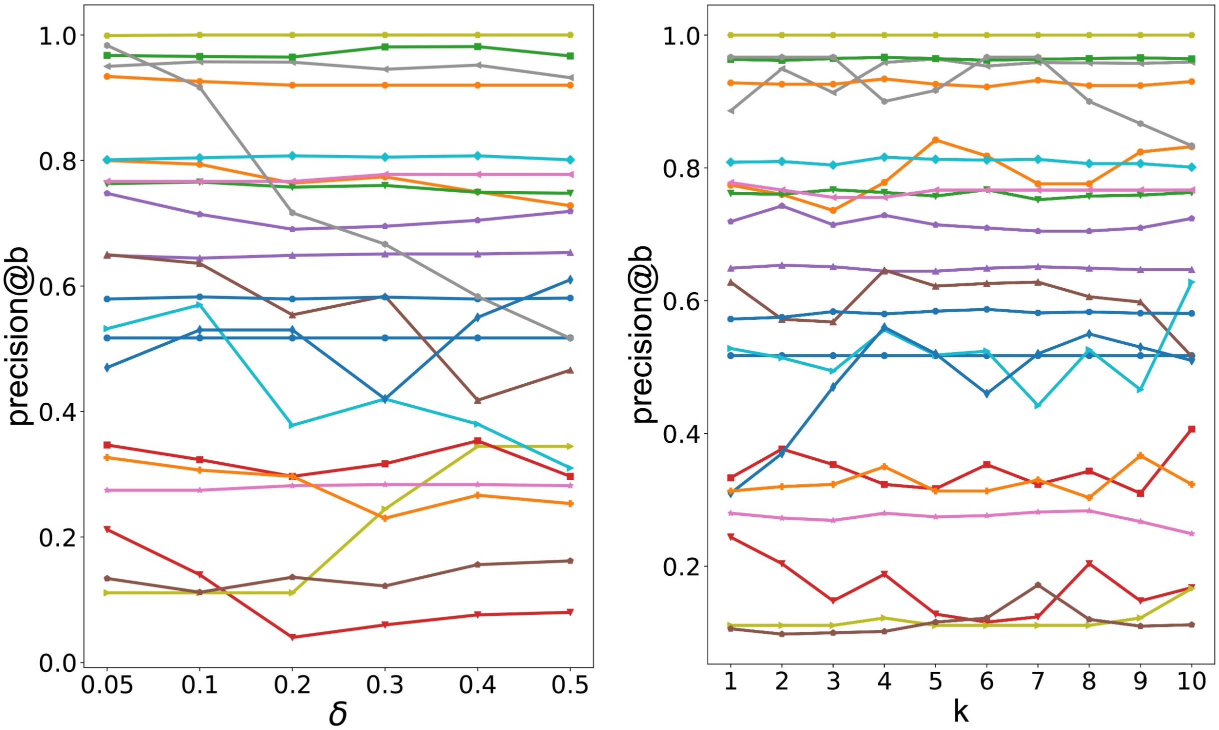


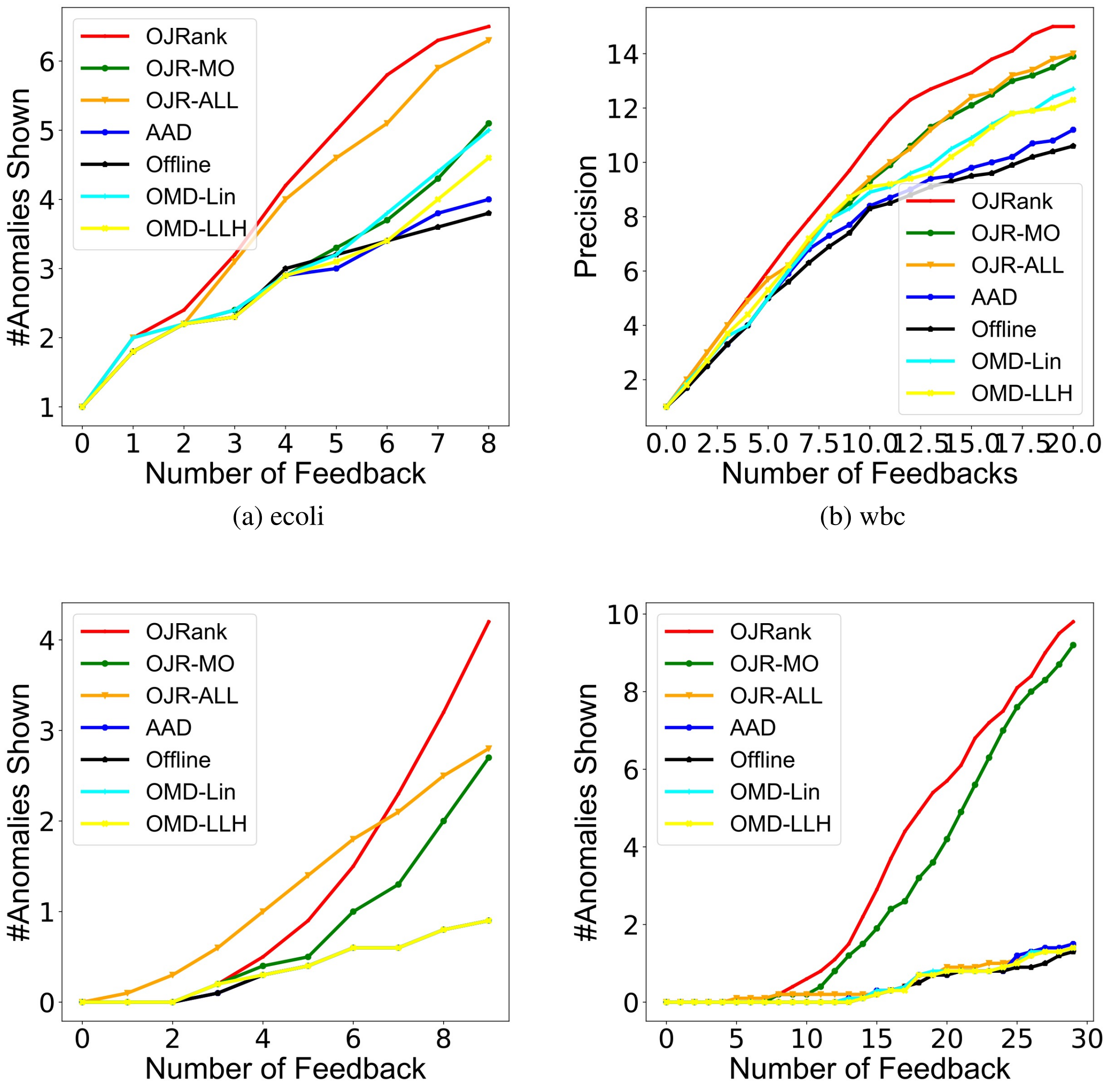
Figure 12.3: *precision@b* remains reasonably stable upon varying (left) δ and (right) k (2 input parameters to OJRANK). Each line corresponds to one of all 14+8 datasets in Table 12.2.

The precision magnitudes differ quite a bit among datasets, therefore we perform a *rank* test to compare the methods statistically. Specifically, the Wilcoxon signed rank test between OJRANK and each baseline shows that OJRANK significantly outperforms its two variants and the offline baseline at p<0.01 on both BENCHMARK and CLUSTERED datasets. (Actual p-values can be found in Figure 12.2 (b).) In fact, notice that Offline is ranked at the bottom in both setups, demonstrating the value of learning on-the-job. OJRANK is also superior to AAD, respectively at p<0.05 and p<0.01. We find no significant difference (p=0.5) between OMD variants and OJRANK on BENCHMARK DATASETS. On the other hand, OJRANK significantly outperforms all baselines including OMD on CLUSTERED DATASETS, showcasing its ability to learn from feedback on clustered instances.

We illustrate how the number of true discovered anomalies change over rounds with the expert on several datasets from BENCHMARK DATASETS and CLUSTERED DATASETS in Figure 12.4 and Figure 12.5, respectively.

***Expert effort*:** Next we analyze the results on *expert effort* on BENCHMARK DATASETS in Table 12.6 and on CLUSTERED DATASETS in Table 12.7. The differences between OJRANK and baselines become apparent especially on this metric. Notice that OJRANK yields significantly better *expert effort* than all of the baselines at p¡0.05. (See Figure 12.2 (b) for the actual p-values.)

As given in Defn. 12.3, *expert effort* utilizes similarity in the anomaly scoring space $\mathcal{S}$. Recall that one could also argue for similarity in the original input space $\mathcal{X}$. To this end, we also report (only) the average rank (for brevity) per method across all datasets based on effort utilizing similarity in $\mathcal{X}$ space, shown in the last row of Tables 12.6 and 12.7. Here, we observe



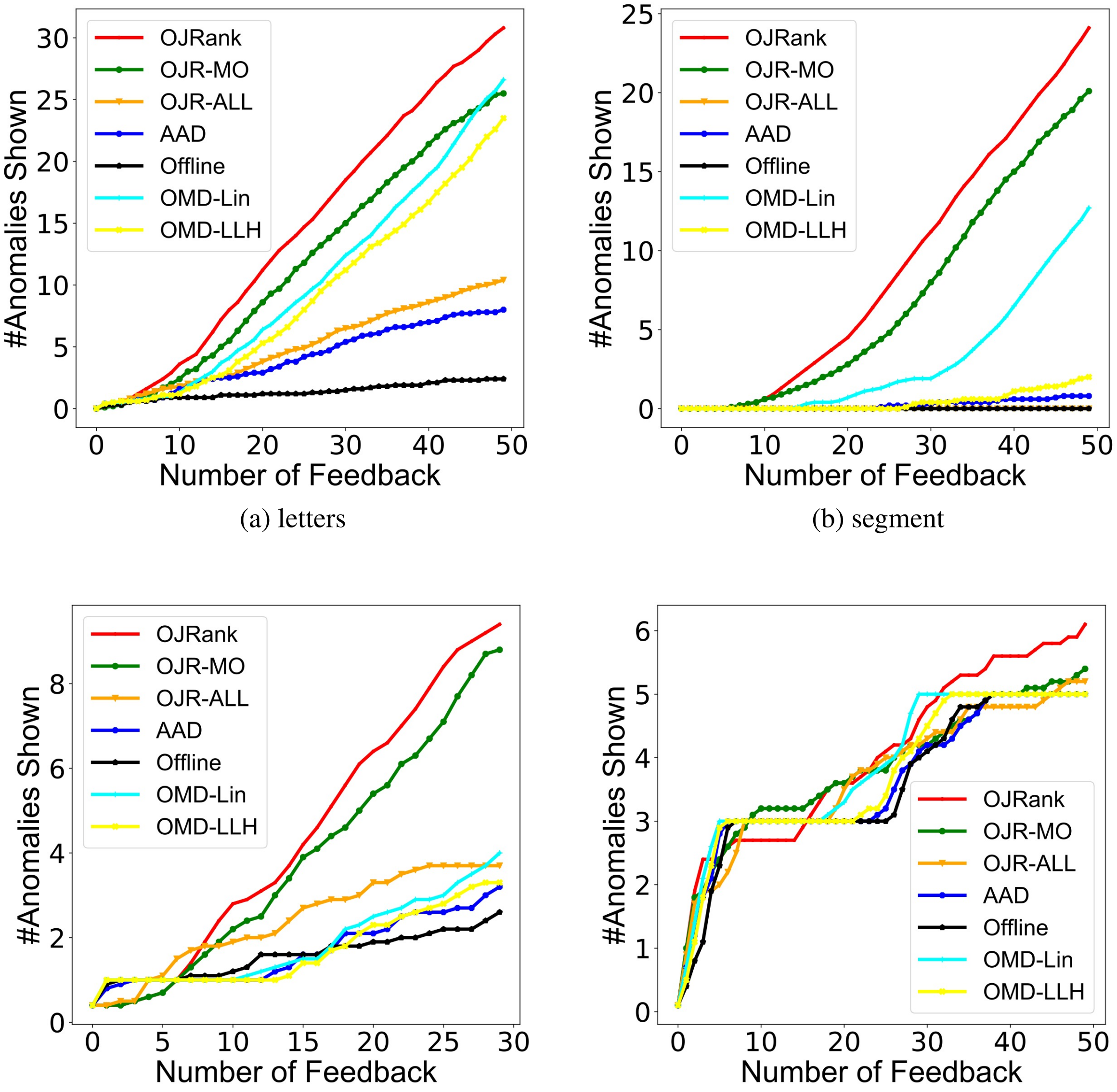
(a) ecoli

(b) wbc

(c) wine

(d) vertebral

Figure 12.4: Number of anomalies shown by each method over feedback rounds for several BENCHMARK DATASETS.



(a) letters

(b) segment

(c) vehicle

(d) svmguide

Figure 12.5: Number of anomalies shown by each method over feedback rounds for several CLUSTERED DATASETS.

the same trends on BENCHMARK DATASETS. OJRank also outperforms all the baselines on CLUSTERED DATASETS at (a slightly higher) p<0.1. The somewhat better effort the baselines achieve in this setup is because they show consecutive instances from the "rare nominal" cluster or from the same larger nominal clusters. This querying of similar instances achieves reduced effort, however at the expense of poor precision (as observed from Table 12.4).

Overall comparison: The ideal method for on-the-job re-ranking is the one that achieves high precision and enables low effort at the same time. To compare all the methods in both grounds, Figure 12.2 (a) presents a scatter plot of the avg. rank w.r.t. *precision@b* versus avg. rank w.r.t. *expert effort* for each setup. It is easy to see that OJRank is closest to the top (denoted by the target symbol on the plots), especially on CLUSTERED DATASETS, where the differences are significant as discussed in the previous subsections.

Response time to update: An important requirement for the kind of applications considered in this work is fast response time; since the expert is to wait between feedbacks to be presented with the updated top-1 instance. In Figure 12.6, we show the distribution of per-round update time (avg.'ed over 10 iForests) over all rounds with boxplots. For brevity, results for a subset of BENCHMARK DATASETS are shown. Moreover, only the state-of-the-art baselines (AAD and OMD) and OJRank are compared.

The key take-aways are two: OJRank takes less than one fifth of a second to provide a model update on average – which would be near instantaneous for a human expert. In addition, the update time has low variance from round to round and from dataset to dataset (unlike e.g., AAD).

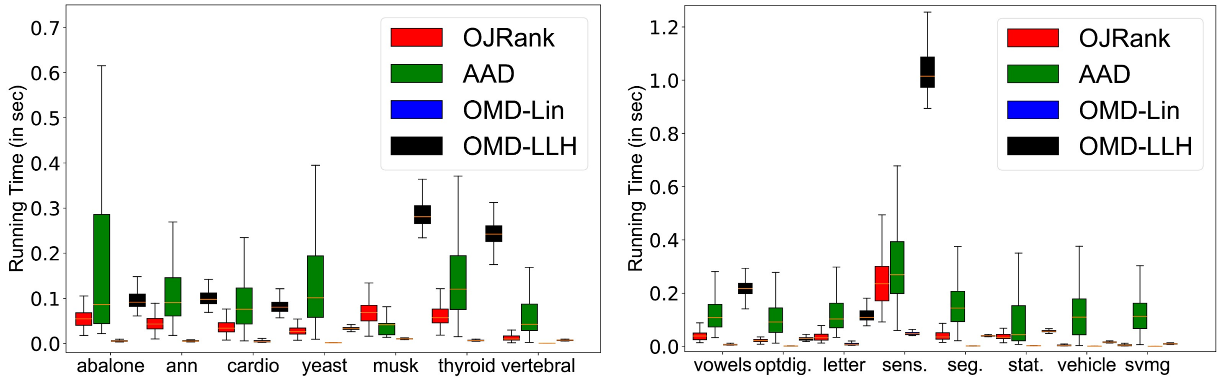


Figure 12.6: Avg. runtime per update on several (left) BENCHMARK & (right) CLUSTERED datasets. OJRank's response time is less than one fifth of a second, with low variance.

12.4.4 Sensitivity Analysis

We conclude experiments with an analysis of OJRank's sensitivity to its input parameters; scaling factor δ and number of pairs k to sample. We tested over small values of the parameters so as not to stray far away from the original ranked list upon a single feedback. As shown in Figure 12.3, performance remains nearly stable for most datasets. We use and recommend $\delta = 0.1$ and $k = 5$.

12.5 Conclusion

In this work we addressed the problem of how to leverage the label revealed by an expert on the top-1 instance to quickly re-rank the anomalies in an online fashion. The proposed approach OJRANK works alongside the expert and continues to learn on-the-job from every top-1 feedback. To this end, OJRANK leverages a cross entropy based pairwise learning to rank objective along with accelerated online gradient updates. These updates correspond to a 'more-like-this' strategy on true positive feedback – boosting other similar instances up the list, and a 'less-like-this' strategy on false positive feedback – muting other similar false positives. We show that OJRANK not only increases *precision* but also decreases *expert effort* over two different classes of datasets, and significantly outperforms the offline and state-of-the-art baselines over both metrics. Finally, OJRANK has constant time complexity with instantaneous response time to update, and linear space requirement on the number of instances.

Part VI

Conclusions and Future Work

CHAPTER 13

CONCLUSIONS

In this thesis, we worked on problems related to characterizing and modeling large scale user behavior on social and technical platforms in aim to discover interesting patterns and find anomalous cases. The works contributing to this thesis can be broadly categorized into two sub-topics: (1) characterizing and modeling user behavior, and (2) anomaly detection. We study human behavior on large scale online social media platforms from two perspectives - the first perspective is based on characterizing interesting phenomena observed on online platforms. In such phenomena, the entity of interest is the phenomena itself, which is a resultant of behavior of multiple users. In contrast, the second perspective we study is about modeling behavior at an individual level. In such problems, the entity of interest is user, and we cover characteristics and model each user's behavior.

13.1 Contributions

13.1.1 User-based Phenomena on Social Media

- **Understanding firestorms**: Chapter 3 measures the effect of firestorms on the attacked entity. The work bases hypothesis on famous sociologicial theory - *biographical consequences of activism*. We operationalize the theory by comparing the mention network between the users participating in firestorms. We discover that mention networks before and after the firestorms are very similar, and they are highly distinct from the week of the firestorm; and this was something further corroborated by network-level statistics. We performed this for 20 firestorm events.
- **Bias in Geocoded tweets**: In Chapter 4, we study the bias that might exist in geotagged content posted on online social media platforms. We analyze 144 million geotagged tweets and link them with high resolution Census population data. We use spatial models to analyze various demographic factors that contribute to geocoded content. We discover that with the use of spatial models, we are able to explain 42 percent of the variance in the data. We observed that the actual population does not have an effect on number of geotagged

users, but having higher median income, being in an urban area, having more young people and having high Asian, Black, and Latino population does.

13.1.2 Individual User Modeling

- **Detecting dangerous selfie behavior on social media**: Chapter 5 characterizes the problem of voluntary risk-taking on online social media platforms. In this work, we specifically studied the behavior of posting dangerous selfies. We propose a multimodal classifier that takes into account three classes of features - location, image and text features to successfully identify if the given selfie posted is dangerous or not. We were able to report 82% accuracy, outperforming other unimodal feature based classifiers.
- **Detecting distracted driving posts on social media**: In Chapter 6, we continue studying risk-taking behavior on social media websites in the form of distracted driving content posted. We propose a deep learning classifier to detect distracted driving content, and were able to achieve 94% accuracy. Further, we tested previously limitedly tested sociological theory, *edgework* framework describing characteristics of voluntary risk takers at a large scale. We conclude that the theory still holds - young males are more likely to post distracted driving content.
- **Modeling experience in recommendation systems**: Chapter 7 focusses on modeling experience of users in recommender systems. The work is hypothesized on that as user interacts with the recommendation system, their preferences evolved and become more nuanced. The changed preferences can be considered as experience. We propose a supervised graphical model which models user preferences over time using the ratings and reviews. We show that the method had a mean square error of 0.363, outperforming other baseline models. We also showed the efficacy of the approach on multiple e-commerce datasets such as BeerAdvocate, RateBeer, Amazon and Yelp.

13.1.3 Identifying Fraud on Social Media

- **Understanding Link Fraud Services**:In Chapter 8, we characterize the multi-faceted nature of link fraud on Twitter. We bought followers from multiple follower delivery services and analyzed the varied, heterogeneous nature of followers so delivered. Additionally, we proposed entropy based machine learning classifier to identify customers who engage in link fraud.
- **Modeling Dwell Time Engagement Fraud**: Chapter 9 focussed on modeling dwell time behavior on visual multimedia content posted on online social platforms. We propose parametric, interpretable models for modeling individual and joint models of dwell time behavior. Furthermore, we also use these models for anomaly detection.
- **Detecting Chatbots on Livestreaming applications**: In Chapter 10, we propose models to detect chatbots on livestreaming platforms. We propose a 2-stage classifer where in the first step we identify chatbotted livestreams, and then in the next step we detect constituent chatbots. We showed that the proposed method is robust under different chatbot attack models and outperforms baselines.

13.1.4 Anomaly Detection Beyond Social Media

- **Individual metrics for group-based temporal fraud**: We propose a new metric for scoring individual entities participating in group-based fraud attacks in Chapter 11. Our contributions encompass two things - (1) leveraging temporal patterns to enhance group level fraud detection and (2) metric for counting each individual's contribution to a group level attack. We were able to show the efficacy of the method for different online social media platforms.
- **Incorporating human feedback for anomaly detection**: In Chapter 12, we propose a novel way of incorporating expert's feedback to re-rank list of outliers. We propose a learning to rank based method which reduces false positive rate of the anomaly detection model and also decreases expert's effort in verifying the anomalies.

13.2 Impact

Besides the above contribution, I below list the notable academic and press-related impact of the works presented in this document.

13.2.1 Academic Impact

- OJRANK (Chapter 12) won the Best Research Paper award at SDM, 2019.
- Work presented in Chapter 3 won the Best Student Paper award at ASONAM, 2015.
- OJRANK (Chapter 12) was featured in KDD 2019 tutorial on rare category exploration.
- Precursor to the work presented in Chapter 4[159] was runner up for SBP Data Challenge. The work itself is one of the top cited papers in ICWSM 2015, having more than 80 citations.

13.2.2 Practical Impact

- Work presented in Chapter 3 was covered by Pittsburgh Post Gazette.
- Work presented in Chapter 5 has been covered by more than 100 media outlets.
- Work presented in Chapter 5 has been presented at multiple universities, and also contributes significantly to a TEDx talk on the same topic.
- ZOORANK (Chapter 11) has been downloaded 1.9K times, and was mentioned in keynote at HotSOS 2016.

CHAPTER 14 FUTURE WORK

The work presented in this thesis is preliminary step of utilizing large dataset presented by online social platforms to find interesting patterns and develop user behavior models. Building on the success of thesis, there are three different streams of future work that could be seen as next logical step for this dissertation. The first stream focuses on stream of computational social science, and involves studying online social platforms from sociological perspective and argue about the role they play in our society. The next stream is centered on cybersecurity applications, and involves developing robust algorithms for fraud detection on social media that can handle both evolving and adversarial nature of fraud. Finally, we hope to extend previously developed methods and also develop new methods for analyzing large scale archival data that exists in other interesting domains.

14.1 Computational Social Science

Building on the work in [160, 161, 188], a promising direction is to continue analyzing large-scale online platform data, model user behavior and, through the lens of these social and technical platforms argue about their role in human society. Specifically, I want to look at the following directions:

14.1.1 Causal Inference

Most social-media-based predictive studies are only focused on correlation. However, given that the implication of these studies are huge and could potentially impact society at large, it is necessary to argue about whether the factors discovered to predict the behavior are causal or not. One of the promising direction lies in studying voluntary risk-taking behavior. Specifically, investigating the impact of social feedback on the intensity of participation in risk-taking activities.

14.1.2 Polarization and Social Media

The role of social media platforms in the recent US elections and other political events has been under scrutiny. One aspect of that discussion has been on the role of echo chambers, and thus polarization. In the current literature, there are two opposing schools of thought - one that postulates that social media platforms increase polarization through echo chambers; and another that argues that these platforms decrease polarization by allowing cross interaction between different ideologies. How can we characterize the role of online social media platforms in increasing or decreasing polarization? How can we model user's change in behavior after interacting with the platform itself?

14.2 CyberSecurity

While already impactful, my work on modeling abnormal dwell time engagement and detecting fraudulent users and reviews and ratings on e-commerce systems is only the first step. There are many more social platforms, where fraudsters have very different modus operandi to cheat the platform for their personal gains. Furthermore, the field of cybersecurity is ever-evolving, with fraudsters often improving their evasion techniques to fool the detection algorithm. Thus, it is of utmost importance to researchers to keep on developing new techniques to detect newer type of fraud. Specifically, I am excited to work on the following topics:

14.2.1 Adversarial Data Mining

In the domain of cybersecurity, as mentioned earlier, adversaries continuously adapt their behavior to evade data mining models. Existing detection methods cannot typically adapt to these changes. One theme of research I will pursue in the future is developing robust adversary-sensitive data mining algorithms.

14.2.2 Human-in-the-loop anomaly detection

: One critical limitation of existing machine-learning-based fraud detection methods is the absence of a human in the loop - the entire framework ends at an output of instances sorted by their outlier score. However, the process requires more scrutiny, since applications typically have such high impact that they require expert verification. In prior work, I proposed a framework that takes into account the expert feedback on highly scored anomalies, and used that to reduce the false positive rate of the anomaly detector as well as the expert's effort. There are multiple possible extensions to the existing framework, which I will explore in the future - e.g. extending the current case of individual anomaly detection to the case of group anomalies and the case of multiple annotators.

14.3 Extending to other domains

Finally, one of my goals is to expand my research into new domains. As described above, analyzing large-scale archival data from online communities, while building on socio-cognitive theories, allow us to model novel behavioral patterns and answer interesting questions. As such, there are fascinating opportunities in software engineering, healthcare, ridesharing gig economy, non-profits, etc. One of the natural extension is to study software engineering domain, by analyzing data available through GitHub, and other code sharing websites. Yet, another interesting domain is to consider gig working platforms and introduce how can we better allocate gigs to ensure fairness among the gig-workers. Collaborating with practitioners in these various domains can enable us to answer these questions effectively.

BIBLIOGRAPHY

[1] Ellen degeneres orchestrates the most famous selfie ever at the oscars. http://www.huffingtonpost.com/2014/03/02/ellen-degeneres-selfie-oscars. Accessed: 2016-05-30.

[2] 24 billion selfies uploaded to google photos in one year. https://googleblog.blogspot.in/2016/05/google-photos-one-year-200-million.html. Accessed: 2016-05-30.

[3] The oxford dictionaries word of the year 2013 is ‘selfie’. http://blog.oxforddictionaries.com/2013/11/word-of-the-year-2013-winner/.

[4] More than half of millennials have shared a selfie. http://www.pewresearch.org/fact-tank/2014/03/04/more-than-half-of-millennials-have-shared-a-selfie/.

[5] Definition of selfie. https://en.oxforddictionaries.com/definition/selfie. Online.

[6] More people have died by taking selfies this year than by shark attacks. http://www.telegraph.co.uk/technology/11881900/More-people-have-died-by-taking-selfies-this-year-than-by-shark-attacks.html.

[7] Google maps api. https://developers.google.com/maps/, 2016.

[8] Central Intelligence Agency. *The CIA world factbook 2010*. Skyhorse Publishing Inc., 2009.

[9] Anupama Aggarwal and Ponnurangam Kumarguru. What they do in shadows: Twitter underground follower market. In *PST*, 2015.

[10] Nancy R Ahern, Penny Sauer, and Paige Thacker. Risky behaviors and social networking sites: how is youtube influencing our youth? *Journal of psychosocial nursing and mental health services*, 53(10):25–29, 2015.

[11] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*, pages 410–421. Springer, 2010.

[12] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. *International Conference on Web and Social Media*, 2013.

[13] American Community Survey Office. ACS summary file technical documentation: 2013 ACS 1-year, 2011-2013 ACS 3-year, and 2009-2013 ACS 5-year data releases. Technical report, 2014. URL http://www2.census.gov/acs2013_5yr/summaryfile/ACS_2013_SF_Tech_Doc.pdf.

[14] Luc Anselin. Under the hood: Issues in the specification and interpretation of spatial regression models. *Agricultural Economics*, 27(3):247–267, 2002.

[15] Luc Anselin and Serge Rey. Properties of tests for spatial dependence in linear regression models. *Geographical Analysis*, 23(2):112–131, 1991. ISSN 1538-4632. doi: 10.1111/j.1538-4632.1991.tb00228.x.

[16] Luc Anselin, Sanjeev Sridharan, and Susan Gholston. Using exploratory spatial data analysis to leverage social indicator databases: The discovery of interesting patterns. *Social Indicators Research*, 82(2):287–309, 2007. ISSN 0303-8300.

[17] Andrew H Avery, Lisa Rae, J Blair Summitt, and Steven Alexander Kahn. The fire challenge: a case report and analysis of self-inflicted flame injury posted on social media. *Journal of Burn Care & Research*, 37(2):e161–e165, 2016.

[18] Anirban K Baishya. #Namo: The political work of the selfie in the 2014 inter-annotatorian general elections. *International Journal of Communication*, 2015.

[19] V Barnett and T Lewis. *Outliers in statistical data*. Wiley, 1994.

[20] Fabrcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virglio Almeida. Detecting spammers on twitter. In *CEAS*, 2010.

[21] Steve Bennett. Log-logistic regression models for survival data. *Applied Statistics*, 1983.

[22] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, 2013.

[23] Sandeep Bhogesha, Jerry R John, and Satyaswarup Tripathy. Death in a flash: selfie and the lack of self-awareness. *Journal of Travel Medicine*, 2016.

[24] Roger Bivand and Gianfranco Piras. Comparing implementations of estimation methods for spatial econometrics. *Journal of Statistical Software*, 63(18):1–36, 2015.

[25] Roger Bivand, Jan Hauke, and Tomasz Kossowski. Computing the Jacobian in Gaussian spatial autoregressive models: An illustrated comparison of available methods. *Geographical Analysis*, 45(2):150–179, 2013. URL http://www.jstatsoft.org/v63/i18/.

[26] Roger S. Bivand, Edzer Pebesma, and Virgilio Gómez-Rubio. *Applied spatial data analysis with R, Second edition*. Springer, NY, 2013.

[27] David M. Blei and Jon D. McAuliffe. Supervised topic models. In *NIPS*, 2007. URL http://books.nips.cc/papers/files/nips20/NIPS2007_0893.pdf.

[28] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, 2003.

[29] YouTube Offical Blog. You know what‘s cool? a billion hours. https://goo.gl/

zPNNT7, 2017.

[30] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. A context-aware time model for web search. In *SIGIR*, 2016.

[31] Léon Bottou. Stochastic learning. In *Summer School on Machine Learning*, pages 146–168. Springer, 2003.

[32] Eric Bouyé, Valdo Durrleman, Ashkan Nikeghbali, Gaël Riboulet, and Thierry Roncalli. Copulas for finance-a reading guide and some applications. *Available at SSRN 1032533*, 2000.

[33] danah boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter. In *HICSS '10: Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, 2010.

[34] Jenna Brager. The selfie and the other: consuming viral tragedy and social media (after) lives. *International Journal of communication*, 2015.

[35] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, 2000.

[36] Axel Bruns and Hallvard Moe. Structural layers of communication on Twitter. In Katrin Weller, Axel Bruns, Jean Burgess, Merja Mahrt, and Cornelius Puschmann, editors, *Twitter and society*, pages 15–28. Peter Lang, 2014.

[37] Axel Bruns and Stefan Stieglitz. Quantitative approaches to comparing communication patterns on Twitter. *Journal of Technology in Human Services*, 30(3–4):160–185, 2012.

[38] Zhuhua Cai and Christopher Jermaine. The latent community model for detecting Sybils in social networks. In *NDSS*, February 2012.

[39] Jeff K Caird, Chelsea R Willness, Piers Steel, and Chip Scialfa. A meta-analysis of the effects of cell phones on driver performance. *Accident Analysis & Prevention*, 40(4): 1282–1293, 2008.

[40] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *CCS*, pages 477–488. ACM, 2014.

[41] Douglas M Carmean and Margaret E Morris. Selfie examinations? applying computer vision, hashtag scraping and sentiment analysis to finding and interpreting selfies, 2015.

[42] David Carr. Hashtag activism, and its limits. *New York Times*, March 2012.

[43] Stevie Chancellor, Yannis Kalantidis, Jessica A. Pater, Munmun De Choudhury, and David A. Shamma. Multimodal classification of moderated online pro-eating disorder content. In *CHI*, 2017.

[44] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*. Springer, 2000.

[45] Sougata Chaudhuri and Ambuj Tewari. Online learning to rank with feedback at the top. In *AISTATS*, pages 277–285, 2016.

[46] Sougata Chaudhuri and Ambuj Tewari. Online ranking with top-1 feedback. In *AISTATS*, pages 129–137, 2015.

[47] Liang Chen, Yipeng Zhou, and Dah Ming Chiu. Analysis and detection of fake views in online video services. *TOMM*, 11(2s), 2015.

[48] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD 2011*, pages 1082–1090, 2011. ISBN 978-1-4503-0813-7.

[49] Eric Chow. Crowd culture & community interaction on twitch.tv. *OSUVA Open Science*, 2016.

[50] Henrik Christensen. Political activities on the internet: Slacktivism or political participation by other means? *First Monday*, 16(2), 2011.

[51] Meredith D. Clark. *To tweet our own cause: A mixed-methods study of the online phenomenon "Black Twitter"*. PhD thesis, The University of North Carolina at Chapel Hill, School of Journalism and Mass Communication, 2014.

[52] Jeremy W. Crampton, Mark Graham, Ate Poorthuis, Taylor Shelton, Monica Stephens, Matthew W. Wilson, and Matthew Zook. Beyond the geotag: Situating "big data" and leveraging the potential of the geoweb. *Cartography and Geographic Information Science*, 40(2):130–139, 2013. doi: 10.1080/15230406.2013.777137.

[53] Claudia Czado. Pair-copula constructions of multivariate copulas. In *Copula theory and its applications*, pages 93–109. Springer, 2010.

[54] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. No country for old members: User lifecycle and linguistic change in online communities. In *WWW*, 2013.

[55] Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott. Incorporating expert feedback into active anomaly discovery. In *IEEE ICDM*, pages 853–858, 2016.

[56] Shubhomoy Das, Weng-Keen Wong, Alan Fern, Thomas G Dietterich, and Md Amran Siddiqui. Incorporating feedback into tree-based anomaly detection. *arXiv preprint arXiv:1708.09441*, 2017.

[57] Arabin Kumar Dey and Debasis Kundu. Discriminating between the log-normal and log-logistic distributions. *Commun. Stat. Theory Methods*, 2009.

[58] Matthew DiPietro. *On Artifical viewers, Followers, and Chat Activity*, 2016.

[59] J. Donahue, Yangqing J., Vinyals O., J. Hoffman, N. Zhang, E. Tzeng, and T. Dravell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[60] Judith Donath. Signals in social supernets. *Journal of Computer-Mediated Communication*, 13(1):231–251, 2007. ISSN 1083-6101.

[61] Pinar Donmez and Jaime G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *ICML*, pages 248–255, 2008.

[62] Derek Doran, Swapna Gokhale, and Aldo Dagnino. Human sensing for smart cities. In *ASONAM '13*, pages 1323–1330, 2013. ISBN 978-1-4503-2240-9.

[63] James A Doyle. *The male experience*. Brown & Benchmark, 1995.

[64] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *NIPS*, 1997.

[65] William Eberle and Lawrence Holder. Discovering structural anomalies in graph-based data. In *ICDMW*, 2007.

[66] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *EMNLP '10*, pages 1277–1287, 2010.

[67] A. F. Costa, Y. Yamaguchi, A. J. M. Traina, C. Traina, Jr., and C. Faloutsos. Rsc: Mining and modeling temporal activity in social media. In *KDD*, pages 269–278. ACM, 2015.

[68] David Fan. Tweets are not public opinion but can be used to predict public opinion. Working papers, University of Minnesota, 2012.

[69] A. C. Favre, S. El Adlouni, L. Perreault, N. Thiémonge, and B. Bobée. Multivariate hydrological frequency analysis using copulas. *Water Resources*, 40(1), 2004.

[70] Alceu Ferraz Costa, Yuto Yamaguchi, Agma Juci Machado Traina, Caetano Traina, Jr., and Christos Faloutsos. Rsc: Mining and modeling temporal activity in social media. In *KDD*, 2015.

[71] Gerard T Flaherty and Joonkoo Choi. The selfie phenomenon: reducing the risk of harm while using smartphones during international travel. *Journal of travel medicine*, 2016.

[72] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 1971.

[73] Joseph L Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973.

[74] Sarah Florini. Tweets, tweeps, and signifyin': Communication and cultural performance on "Black Twitter". *Television & New Media*, 15(3):223–237, 2014. doi: 10.1177/1527476413480247.

[75] David Mandell Freeman. Using naive bayes to detect spammy names in social networks. In *AISec*, pages 3–12. ACM, 2013.

[76] V. Frias-Martinez, V. Soto, H. Hohwald, and E. Frias-Martinez. Characterizing urban landscapes using geolocated tweets. In *PASSAT/SocialCom '12*, pages 239–248, Sept 2012. doi: 10.1109/SocialCom-PASSAT.2012.19.

[77] Maksym Gabielkov and Arnaud Legout. The complete picture of the Twitter social graph. In *CoNEXT Student '12: Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop*, pages 19–20, 2012.

[78] Carlo Gaetan and Xavier Guyon. *Spatial Statistics and Modeling*. Springer Series in Statistics. Springer, 2012.

[79] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *SIGCOMM*, 2010.

[80] Rayid Ghani and Mohit Kumar. Interactive learning for efficiently detecting errors in

insurance claims. In *KDD*, pages 325–333. ACM, 2011.

[81] Shayan Oveis Gharan, Farnaz Ronaghi, and Ying Wang. What memes say about the news cycle. Technical report, Tech. rep., Stanford University, 2010.

[82] Debarchana (Debs) Ghosh and Rajarshi Guha. What are we tweeting about obesity?: Mapping tweets with topic modeling and geographic information system. *Cartography and Geographic Information Science*, 40(2):90–102, 2013. doi: 10.1080/15230406.2013.776210.

[83] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the Twitter social network. In *WWW '12: Proceedings of the 21st International Conference on World Wide Web*, pages 61–70, 2012.

[84] Steven Gianvecchio, Mengjun Xie, Zhenyu Wu, and Haining Wang. Humans and bots in internet chat: measurement, analysis, and automated classification. *IEEE/ACM Transactions On Networking*, 2011.

[85] GlobalWebIndex. Social report https://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2017-reports/2016-nielsen-social-media-report.pdf, 2018.

[86] Erving Goffman. The presentation of self in everyday life. *Butler, Bodies that Matter*, 1959.

[87] Andrew V Goldberg. *Finding a maximum density subgraph*. Technical Report, 1984.

[88] Scott A Golder and Michael W Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

[89] Scott A Golder, Dennis M Wilkinson, and Bernardo A Huberman. Rhythms of social interaction: Messaging within a massive online network. In *Communities and technologies 2007*, pages 41–66. Springer, 2007.

[90] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *J. Artif. Intell. Res. (JAIR)*, 46:235–262, 2013.

[91] Mark Graham, Scott A. Hale, and Devin Gaffney. Where in the world are you?: Geolocation and language identification in Twitter. *The Professional Geographer*, 66(4):568–578, 2014.

[92] Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1:201–233, 1983.

[93] Amelia Grant-Alfieri, Judy Schaechter, and Steven E Lipshultz. Ingesting and aspirating dry cinnamon by children and adolescents: the cinnamon challenge. *Pediatrics*, 131(5):833–835, 2013.

[94] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *NIPS*, pages 513–520, 2007.

[95] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @ spam: the underground on 140 characters or less. In *CCS*, pages 27–37. ACM, 2010.

[96] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. Technical report, University of Massachusetts, Amherst, 2002.

[97] Nir Grinberg, Mor Naaman, Blake Shaw, and Gilad Lotan. Extracting diurnal patterns of real world activity from social media. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[98] DJ Guan, Chia-Mei Chen, and Jia-Bin Lin. Anomaly based malicious url detection in instant messaging. In *Proceedings of the joint workshop on information security (JWIS)*, 2009.

[99] Guardian. A selfie with a weapon kills': Russia launches campaign urging photo safety. https://www.theguardian.com/world/2015/jul/07/a-selfie-with-a-weapon-kills-russia-launches-safe-selfie-campaign, 2015.

[100] Jennifer Guay. Most dangerous selfies. http://www.dailymail.co.uk/news/article-3690165/The-world-s-dangerous-selfies-meet-adventure-photographers-putting-lives-risk-perfect-self-portrait.html, 2014.

[101] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[102] Sudipto Guha et al. Robust random cut forest based anomaly detection on streams. In *ICML*, 2016.

[103] Stephan Günnemann, Nikou Günnemann, and Christos Faloutsos. Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution. In *KDD*, 2014.

[104] Diansheng Guo and Chao Chen. Detecting non-personal and spam users on geo-tagged Twitter network. *Transactions in GIS*, 18(3), 2014.

[105] Aditi Gupta, Hemank Lamba, and Ponnurangam Kumaraguru. $1.00 per rt #bostonmarathon #prayforboston: Analyzing fake content on twitter. In *eCRS*, 2013.

[106] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy. In *WWW*, 2013.

[107] Charles N Haas, Josh Joffe, Mark S Heath, and Joseph Jacangelo. Continuous flow residence time distribution function characterization. *Journal of Environmental Engineering*, 123, 1997.

[108] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton Univ. Press, 1994.

[109] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pages 18–22, 2018.

[110] Eszter Hargittai and Eden Litt. The tweet smell of celebrity success: Explaining variation in Twitter adoption among a diverse group of young adults. *New Media & Society*, 13(5): 824–842, 2011.

[111] Matt Hart. Being naked on the internet: young peoples selfies as intimate edgework. *Journal of Youth Studies*, 20(3):301–315, 2017.

[112] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[113] Haibo He and Edwardo Garcia, A. Learning from imbalacned data. *IEEE TKDE*, 2009.

[114] Jingrui He and Jaime G. Carbonell. Nearest-neighbor-based active learning for rare category detection. In *NIPS*, pages 633–640, 2007.

[115] Jingrui He and Jaime G. Carbonell. Rare class discovery based on active learning. In *ISAIM*, 2008.

[116] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[117] Brent Hecht and Monica Stephens. A tale of cities: Urban biases in volunteered geographic information. In *ICWSM '14*, pages 197–205, 2014.

[118] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles. In *CHI '11*, pages 237–246, 2011. URL http://doi.acm.org/10.1145/1978942.1978976.

[119] Bill Heil and Mikolaj Piskorski. New Twitter research: Men follow men and nobody tweets. *Harvard Business Review*, 6 2006. URL https://hbr.org/2009/06/new-twitter-research-men-follo/.

[120] Nadav Hochman and Raz Schwartz. Visualizing instagram: Tracing cultural visual rhythms. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[121] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Inf. Retr.*, 16(1):63–90, 2013.

[122] Bernie Hogan. The presentation of self in the age of social media: Distinguishing performances and exhibitions online. *Bulletin of Science, Technology & Society*, 2010.

[123] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the Twitter stream. In *WWW '12*, pages 769–778, 2012. ISBN 978-1-4503-1229-5.

[124] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, 2016.

[125] William J Horrey and Christopher D Wickens. Examining the impact of cell phone conversations on driving using meta-analytic techniques. *Human factors*, 48(1):196–205, 2006.

[126] Matt Howes. Let me take a# selfie: An analysis of how cycling should respond to the increasing threats posed by exuberant spectators? *Laws of the Game*, 2015.

[127] Shreya Jain, Dipankar Niranjan, Hemank Lamba, Neil Shah, and Ponnurangam Kumaraguru. Characterizing and detecting chatbots on livestreaming platforms. In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis*

and Mining, 2019.

[128] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4), 2002.

[129] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we Twitter: Understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007.

[130] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. Catchsync: catching synchronized behavior in large directed graphs. In *KDD*, pages 941–950. ACM, 2014.

[131] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. Inferring strange behavior from connectivity pattern in social networks. In *PAKDD*, pages 126–138. Springer, 2014.

[132] Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. A general suspiciousness metric for dense blocks in multimodal data. In *ICDM*, 2015.

[133] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.

[134] D. Juan, N. Shah, M. Tang, Z. Qian, D. Marculescu, and C. Faloutsos. M3a: Model, metamodel and anomaly detection for inter-arrivals of web searches and postings. In *DSAA*, pages 341–350, Oct 2017.

[135] Levente Juhász and Hartwig H Hochmair. Analyzing the spatial and temporal dynamics of snapchat. *AnaLysis, Integration, Vision, Engagement (VGI-ALIVE) Workshop*, 2018.

[136] Krishna Y. Kamath, James Caverlee, Kyumin Lee, and Zhiyuan Cheng. Spatio-temporal dynamics of online memes: A study of geo-tagged tweets. In *WWW '13*, pages 667–678, 2013. ISBN 978-1-4503-2035-1. URL http://dl.acm.org/citation.cfm?id=2488388.2488447.

[137] Laura Kann, Tim McManus, William A Harris, Shari L Shanklin, Katherine H Flint, Joseph Hawkins, Barbara Queen, Richard Lowry, Emily OMalley Olsen, David Chyen, Lisa Whittle, Jemekia Thornton, Connie Lin, Yoshimi Yamakawa, Nancy Berner, and Stephanie Zaza. Youth risk behavior surveillanceunited states, 2015. *MMWR Surveillance Summaries*, 65(6):1–180, 2016.

[138] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[139] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[140] Yoojung Kim, Dongyoung Sohn, and Sejung Marina Choi. Cultural difference in motivations for using social network sites: A comparative study of american and korean college

students. *Computers in human behavior*, 27(1):365–372, 2011.

[141] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM*, pages 193–202. ACM, 2014.

[142] Allan J. Kimmel and Philip J. Kitchen. WOM and social media: Presaging future directions for research and practice. *Journal of Marketing Communications*, 20(1-2):5–20, 2014.

[143] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. "I'm eating a sandwich in Glasgow": Modeling locations with tweets. In *SMUC '11*, pages 61–68, 2011. ISBN 978-1-4503-0949-3. doi: 10.1145/2065023.2065039. URL http://doi.acm.org/10.1145/2065023.2065039.

[144] Lorenz Cuno Klopfenstein, Saverio Delpriori, Silvia Malatini, and Alessandro Bogliolo. The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In *DIS*, 2017.

[145] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *PVLDB*, 1998.

[146] Michael Koliska and Jessica Roberts. Selfies— selfies: Witnessing and participatory journalism with a point of view. *International Journal of Communication*, 9:14, 2015.

[147] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*, 2008.

[148] Yehuda Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4), 2010.

[149] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*. Springer, 2011.

[150] Nina Krüger, Stefan Stieglitz, and Tobias Potthoff. Brand communication in Twitter: A case study on Adidas. In *PACIS 2012: Pacific Asia Conference on Information Systems*, 2012.

[151] Shamanth Kumar, Xia Hu, and Huan Liu. A behavior analytics approach to identifying tweets from crisis regions. In *HyperText*, pages 255–260, 2014. URL http://doi.acm.org/10.1145/2631775.2631814.

[152] Srijan Kumar and Neil Shah. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*, 2018.

[153] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th International Conference on World Wide Web*, pages 591–600, 2010. ISBN 978-1-60558-799-8.

[154] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *Journal of Optimization*, 9(1), 1998.

[155] Himabindu Lakkaraju, Chiranjib Bhattacharyya, Indrajit Bhattacharya, and Srujana Merugu. Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In *SDM*, 2011.

[156] AK Lakshmi. The selfie culture: Narcissism or counter hegemony? *Journal of Communication and media Studies*, 2015.

[157] Hemank Lamba and Leman Akoglu. Learning on-the-job to re-rank anomalies from top-1 feedback. In *SIAM International Conference on Data Mining*, 2019.

[158] Hemank Lamba and Neil Shah. Modeling dwell time engagement on visual multimedia. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1104–1113. ACM, 2019.

[159] Hemank Lamba, Momin M Malik, Constantine Nakos, and Jürgen Pfeffer. Rich people don?t have more followers! overcoming social inequality with social media. In *Proceedings of the International Social Computing, Behavioral-Cultural Modeling and Prediction Conference (SBP15) Grand Data Challenge*, 2015.

[160] Hemank Lamba, Momin M Malik, and Jürgen Pfeffer. A tempest in a teacup? analyzing firestorms on twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 17–24. IEEE, 2015.

[161] Hemank Lamba, Varun Bharadhwaj, Mayank Vachher, Divyansh Agarwal, Megha Arora, Niharika Sachdeva, and Ponnurangam Kumaraguru. From camera to deathbed: Understanding dangerous selfies on social media. In *In Proceedings of AAAI International Conference on Weblogs and Social Media*, 2017.

[162] Hemank Lamba, Thomas J Glazier, Javier Cámara, Bradley Schmerl, David Garlan, and Jürgen Pfeffer. Model-based cluster analysis for identifying suspicious activity sequences in software. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, pages 17–22. ACM, 2017.

[163] Hemank Lamba, Bryan Hooi, Kijung Shin, Christos Faloutsos, and Jürgen Pfeffer. zoorank: Ranking suspicious entities in time-evolving tensors. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 68–84. Springer, 2017.

[164] Hemank. Lamba, Dheeraj Reddy. Srikanth. Shashank, Pailla, Singh Shwetanshu, Karandeep. Juneja, and Ponnurangam Kumaraguru. Driving the last mile: Characterizing and understanding distracted driving posts on social networks. In *International Conference on Web and Social Media*, 2020.

[165] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977.

[166] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *KDD*, pages 157–166. ACM, 2005.

[167] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[168] Mark R Leary, Tchividjian LR, and Kraxberger BE. Self-presentation can be hazardous to your health: impression management and health risk. *Health Psychology*, 1994.

[169] Jay Yoon Lee, U Kang, Danai Koutra, and Christos Faloutsos. Fast anomaly detection despite the duplicates. In *WWW Companion*, 2013.

[170] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. In *SIGIR*, pages 435–442. ACM, 2010.

[171] Kalev Leetaru, Shaowen Wang, Guofeng Cao, Anand Padmanabhan, and Eric Shook. Mapping the global Twitter heartbeat: The geography of Twitter. *First Monday*, 18(5), 2013. ISSN 13960466. URL http://firstmonday.org/ojs/index.php/fm/article/view/4366.

[172] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, 2009.

[173] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Memetracker: tracking news phrase over the web, 2009.

[174] Weng Marc Lim and Jonathan Schroeder. Understanding the selfie phenomenon: current insights and future research directions. *European Journal of Marketing*, 50(9/10), 2016.

[175] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, 2009.

[176] Yu-Ru Lin and Drew Margolin. The ripple of fear, sympathy and solidarity during the Boston bombings. *EPJ Data Science*, 3(1):31, 2014. doi: 10.1140/epjds/s13688-014-0031-z. URL http://dx.doi.org/10.1140/epjds/s13688-014-0031-z.

[177] Krsto Lipovac, Miroslav Derić, Milan Tešić, Zoran Andrić, and Bojan Marić. Mobile phone use while driving-literary review. *Transportation research part F: traffic psychology and behaviour*, 47:132–142, 2017.

[178] Chao Liu, Ryen W. White, and Susan Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *SIGIR*, pages 379–386, 2010.

[179] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, pages 413–422, 2008.

[180] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *TKDD*, 6(1):3, 2012.

[181] Yabing Liu, Chloe Kliman-Silver, and Alan Mislove. The tweets they are a-changin: Evolution of Twitter users and behavior. In *ICWSM '14: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[182] Bo Long, Jiang Bian, Olivier Chapelle, Ya Zhang, Yoshiyuki Inagaki, and Yi Chang. Active learning for ranking through expected loss optimization. *IEEE TKDE*, 27(5):1180–1191, 2015.

[183] Paul A. Longley, Muhammad Adnan, and Guy Lansley. The geotemporal demographics of Twitter usage. *Environment and Planning A*, 47(2):465–484, 2015. URL http://www.envplan.com/abstract.cgi?id=a130122p.

[184] Stephen Lyng. Edgework: A social psychological analysis of voluntary risk taking. *American Journal of Sociology*, 95(4):851–886, 1990.

[185] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender

systems with social regularization. In *WSDM*, 2011.

[186] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[187] Momin Malik, Hemank Lamba, Constantine Nakos, and Jürgen Pfeffer. Population bias in geotagged tweets. In *ICWSM '15: Papers from the 2015 ICWSM Workshop on Standards and Practices in Large-Scale Social Media Research*, pages 18–27, 2015.

[188] Momin M Malik, Hemank Lamba, Constantine Nakos, and Jürgen Pfeffer. Population bias in geotagged tweets. *People*, 1(3,759.710):3–759, 2015.

[189] Lev Manovich. Selfie city. http://selfiecity.net/, 2016.

[190] M. Marciel, R. Cuevas, A. Banchs, R. González, S. Traverso, M. Ahmed, and A. Azcorra. Understanding the detection of view fraud in video content portals. In *International Conference on World Wide Web (WWW)*, pages 357– 368. IW3C2, 2016.

[191] Paris Martineau. *Inside YouTube's Fake Views Economy*, 2018.

[192] Doug McAdam. The biographical consequences of activism. *American Sociological Review*, 54(5):744–760, 1989.

[193] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *RecSys*, 2013.

[194] Julian McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW*, 2013.

[195] Michael Mccord and M Chuah. Spam detection on twitter using traditional classifiers. In *ICATC*, pages 175–186. Springer, 2011.

[196] John P McIntire, Lindsey K McIntire, and Paul R Havig. Methods for chatbot detection in distributed text-based communications. In *IEEE International Symposium on Collaborative Technologies and Systems*, 2010.

[197] Sears Merritt, Abigail Jacobs, Winter Mason, and Aaron Clauset. Detecting friendship within dynamic online interaction networks. In *ICWSM '13: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pages 380–389, 2013. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6049.

[198] Rohan Miller and Natalie Lammas. Social media and its implications for viral marketing. *Asia Pacific Public Relations Journal*, 11(1):1–9, 2010.

[199] David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, 2008.

[200] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J. Rosenquist. Understanding the demographics of Twitter users. In *ICWSM '11: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[201] Lewis Mitchell, Morgan R. Frank, Kameron Decker Harris, Peter Sheridan Dodds, and Christopher M. Danforth. The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place. *PLoS ONE*, 8(5):

e64417, 05 2013. doi: 10.1371/journal.pone.0064417.

[202] Anastasia Mochalova and Alexandros Nanopoulos. Restricting the spread of firestorms in social networks. In *ECIS 2014: Twenty Second European Conference on Information Systems*, 2014.

[203] Evgeny Morozov. From slacktivism to activism. *Foreign Policy*, September 2009.

[204] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen Carley. Is the sample good enough? comparing data from Twitter's Streaming API with Twitter's firehose. In *ICWSM '13: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[205] Fred Morstatter, Nichola Lubold, Heather Pon-Barry, Jürgen Pfeffer, and Huan Liu. Finding eyewitness tweets during crises. In *ACL LACSS '14*, pages 23–27, June 2014. URL http://www.aclweb.org/anthology/W/W14/W14-2509.

[206] Fred Morstatter, Jürgen Pfeffer, and Huan Liu. When is it biased?: Assessing the representativeness of Twitter's Streaming API. In *WWW Companion '14: Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pages 555–556, 2014. ISBN 978-1-4503-2745-9. doi: 10.1145/2567948.2576952. URL http://dx.doi.org/10.1145/2567948.2576952.

[207] Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. Re: Captchas: Understanding captcha-solving services in an economic context. In *USENIX Security*, pages 28–28, 2010.

[208] Subhabrata Mukherjee, Gaurab Basu, and Sachindra Joshi. Joint author sentiment topic model. In *SDM*, 2014.

[209] Subhabrata Mukherjee, Hemank Lamba, and Gerhard Weikum. Experience-aware item recommendation in evolving review communities. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 925–930. IEEE, 2015.

[210] Ryan Murphy. The rationality of literal tide pod consumption. *SSRN*, 2018.

[211] José MR Murteira and Óscar D Lourenço. Health care utilization and self-assessed health: specification of bivariate models using copulas. *Empirical Economics*, 41(2), 2011.

[212] Seth A. Myers and Jure Leskovec. The bursty dynamics of the Twitter information network. In *WWW '14: Proceedings of the 23rd International Conference on World Wide Web*, pages 913–924, 2014. ISBN 978-1-4503-2744-2.

[213] Ruchit Nagar, Qingyu Yuan, C. Clark Freifeld, Mauricio Santillana, Aaron Nojima, Rumi Chunara, and S. John Brownstein. A case study of the New York City 2012-2013 influenza season with daily geocoded Twitter data from temporal and spatiotemporal perspectives. *J Med Internet Res*, 16(10), Oct 2014. doi: 10.2196/jmir.3416.

[214] Thomas Nagler. Vinecopula. https://cran.r-project.org/web/packages/VineCopula/VineCopula.pdf, 2018.

[215] Vedant Nanda, Hemank Lamba, Divyansh Agarwal, Megha Arora, Niharika Sachdeva, and Ponnurangam Kumaraguru. Stop the killfies! using deep learning models to identify dangerous selfies. In *Companion of the The Web Conference 2018 on The Web Conference*

2018, pages 1341–1345. International World Wide Web Conferences Steering Committee, 2018.

[216] NCSA. Distracted driving 2015: traffic safety facts research note (rep. no. dot hs 812 381), 2017.

[217] R. B. Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

[218] NHTSA. Policy statement and compiled faqs on distracted driving, 2017.

[219] Nielsen. 2016 nielsen social media report `https://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2017-reports/2016-nielsen-social-media-report.pdf`, 2017. URL `https://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2017-reports/2016-nielsen-social-media-report.pdf`.

[220] Nir Nissim, Aviad Cohen, Robert Moskovitch, Asaf Shabtai, Mattan Edry, Oren Bar-Ad, and Yuval Elovici. Alpd: Active learning framework for enhancing the detection of malicious pdf files. In *JISIC*, pages 91–98. IEEE, 2014.

[221] Tanya Nitins and Jean Burgess. Twitter, brands, and user engagement. In Katrin Weller, Axel Bruns, Jean Burgess, Merja Mahrt, and Cornelius Puschmann, editors, *Twitter and society*, pages 293–304. Peter Lang, 2014.

[222] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.

[223] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, pages 201–210. ACM, 2007.

[224] Jaram Park, Meeyoung Cha, Hoh Kim, and Jaeseung Jeong. Managing bad news in social media: A case study on Domino's pizza crisis. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, pages 282–289, 2012.

[225] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, 2000.

[226] Dan Pelleg and Andrew W. Moore. Active learning for anomaly and rare-category detection. In *NIPS*, pages 1073–1080, 2004.

[227] Nicole Perloth. Fake twitter followers become multimillion-dollar business, April 2013. URL `http://bits.blogs.nytimes.com/2013/04/05/fake-twitter-followers-becomes-multimillion-dollar-business/`. [Online; posted 5-April-2013].

[228] Tom'as Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102 (2):275–304, 2016.

[229] Jürgen Pfeffer, Thomas Zorbach, and Kathleen M. Carley. Understanding online firestorms: Negative word-of-mouth dynamics in social media networks. *Journal of Marketing Communications*, 20(1–2):117–128, 2014.

[230] Karine Pires and Gwendal Simon. Youtube live and twitch: a tour of user-generated live

streaming systems. In *ACM Multimedia Systems Conference*, 2015.

[231] Barbara Poblete, Ruth Garcia, Marcelo Mendoza, and Alejandro Jaimes. Do all birds tweet the same? Characterizing Twitter around the world. In *CIKM '11: Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1025–1030, 2011.

[232] Tichelle Carol-Denise Porch. *Society, Culture, and the Selfie: Analysis of the Impact of the Selfie Practice on Women's Body Image*. PhD thesis, Emory University, 2015.

[233] B Aditya Prakash, Ashwin Sridharan, Mukund Seshadri, Sridhar Machiraju, and Christos Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD*, pages 435–448. Springer, 2010.

[234] Lin et al. Qiu. What does your selfie say about you? *Computers in Human Behavior*, 2015.

[235] Jay Rajasekera. Crisis management in social media and digital age: Recall problem and challenges to toyota. Working papers, Research Institute, International University of Japan, 2010.

[236] Daniel Ramage, Christopher D. Manning, and Susan Dumais. Partially labeled topic models for interpretable text mining. In *KDD*, 2011.

[237] Shebuti Rayana. ODDS library, 2016. URL http://odds.cs.stonybrook.edu.

[238] Martin Ridout, Clarice GB Demétrio, and John Hinde. Models for count data with many zeros. In *IBC*, volume 19, pages 179–192. IBS, 1998.

[239] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on Twitter. In *WWW '11: Proceedings of the 20th International Conference on World Wide Web*, pages 695–704, 2011.

[240] Jon Ronson. How one stupid tweet blew up Justine Sacco's life. *New York Times Magazine*, February 2015. URL http://www.nytimes.com/2015/02/15/magazine/how-one-stupid-tweet-ruined-justine-saccos-life.html.

[241] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[242] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[243] Lauren O Roussel and Derek E Bell. Tweens feel the burn:salt and ice challenge burns. *International journal of adolescent medicine and health*, 28(2):217–219, 2016.

[244] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 1573-1405.

[245] Derek Ruths and Jrgen Pfeffer. Social media for large studies of behavior. *Science*, 346

(6213):1063–1064, 2014.

[246] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860. ACM, 2010.

[247] Takeshi et al. Sakaki. Tweet trend analysis in an emergency situation. In *Proceedings of the Special Workshop on Internet and Disasters*, page 3. ACM, 2011.

[248] Molly Sauter. 'LOIC will tear us apart': The impact of tool design and media portrayals in the success of activist DDOS attacks. *American Behavioral Scientist*, 57(7):983–1007, 2013.

[249] Grant Schoenebeck. Potential networks, contagious communities, and understanding social network structure. In *WWW '13: Proceedings of the 22nd International Conference on World Wide Web*, pages 1123–1132, 2013.

[250] Theresa M Senft and Nancy K Baym. Selfies introduction˜ what does the selfie say? investigating a global phenomenon. *International Journal of Communication*, 9:19, 2015.

[251] Burr Settles. Active learning. *Synthesis Lectures on AI and ML*, 6(1):1–114, 2012.

[252] Neil Shah. Flock: Combating astroturfing on livestreaming platforms. In *International Conference on World Wide Web (WWW)*, pages 1083–1091, 2017.

[253] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *ICDM*, 2014.

[254] Neil Shah, Danai Koutra, Tianmin Zou, Brian Gallagher, and Christos Faloutsos. Timecrunch: Interpretable dynamic graph summarization. In *KDD*. ACM, 2015.

[255] Neil Shah, Hemank Lamba, Alex Beutel, and Christos Faloutsos. The many faces of link fraud. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 1069–1074. IEEE, 2017.

[256] Sanjay Sharma. Black Twitter?: Racial hashtags, networks and contagion. *New Formations: A Journal of Culture/Theory/Politics*, 78(1), 2013. URL http://muse.jhu.edu/journals/new_formations/v078/78.sharma.html.

[257] Taylor Shelton, Ate Poorthuis, Mark Graham, and Matthew Zook. Mapping the data shadows of Hurricane Sandy: Uncovering the sociospatial dimensions of 'big data'. *Geoforum*, 52(0):167 – 179, 2014. URL http://www.sciencedirect.com/science/article/pii/S0016718514000207.

[258] Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. Corescope: Graph mining using k-core analysis - patterns, anomalies and algorithms. In *ICDM*, 2016.

[259] Kijung Shin, Bryan Hooi, and Christos Faloutsos. M-zoom: Fast dense-block detection in tensors with quality guarantees. In *ECML/PKDD*, 2016.

[260] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, Ryan Wright, Alec Theriault, and David W. Archer. Feedback-guided anomaly discovery via online optimization. In *KDD*. ACM, 2018.

[261] Rodrigo M. Silva, Guilherme de Castro Mendes Gomes, Mrio S. Alvim, and Marcos Andr Gonalves. Compression-based selective sampling for learning to rank. In *CIKM*, pages

247–256. ACM, 2016.

[262] Matthew P Simmons, Lada A Adamic, and Eytan Adar. Memes online: Extracted, subtracted, injected, and recollected. In *ICWSM*, 2011.

[263] Snapchat. Snapchat for business (https://forbusiness.snapchat.com), 2018.

[264] Monica Stephens and Ate Poorthuis. Follow thy neighbor: Connecting the social and the spatial networks on Twitter. *Computers, Environment and Urban Systems*, 2014. URL http://www.sciencedirect.com/science/article/pii/S0198971514000726.

[265] Stefan Stieglitz and Nina Krüger. Analysis of sentiments in corporate Twitter communication: A case study on an issue of Toyota. In *ACIS 2011: Australiasian Conference on Information Systems*, 2011.

[266] Stefan Stieglitz and Nina Krüger. Public enterprise-related communication and its impact on social media issue management. In *Twitter and society*, pages 281–292. Peter Lang, 2014.

[267] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *ACSAC*, pages 1–9. ACM, 2010.

[268] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Y. Ben Zhao. Follow the green: Growth and dynamics in twitter follower markets. In *SIGMETRICS*, 2013.

[269] BV Subrahmanyam and et al. Selfie related deaths perils of newer technologies. *Narayana Medical Journal*, 2016.

[270] Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM*, 2005.

[271] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *arXiv preprint arXiv:1906.06821*, 2019.

[272] Jared Sylvester, John Healey, Chen Wang, and William M. Rand. Space, time, and hurricanes: Investigating the spatiotemporal relationship among social media use, donations, and disasters. Technical Report Research Paper No. RHS 2441314, Robert H. Smith School, 2014. URL http://dx.doi.org/10.2139/ssrn.2441314.

[273] Christian et al. Szegedy. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. URL http://arxiv.org/abs/1512.00567.

[274] Yuri Takhteyev, Anatoliy Gruzd, and Barry Wellman. Geography of Twitter networks. *Social Networks*, 34(1):73–81, 2012. ISSN 0378-8733. doi: http://dx.doi.org/10.1016/j.socnet.2011.05.006. URL http://www.sciencedirect.com/science/article/pii/S0378873311000359.

[275] Swee Chuan Tan, Kai Ming Ting, and Fei Tony Liu. Fast anomaly detection for streaming data. In *IJCAI*, pages 1511–1516, 2011.

[276] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of Twitter spam. In *IMC '11: Proceedings of the 2011 ACM SIG-*

COMM Conference on Internet Measurement Conference, pages 243–258, 2011. ISBN 978-1-4503-1013-0.

[277] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking fraudulent accounts: The role of the underground market in Twitter spam and abuse. In *USENIX Security '13: Proceedings of the 22nd USENIX Security Symposium*, pages 195–210, 2013.

[278] Kurt Thomas, Dmytro Iatskiv, Elie Bursztein, Tadek Pietraszek, Chris Grier, and Damon McCoy. Dialing back abuse on phone verified accounts. In *CCS*, 2014.

[279] Robert L Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[280] Alise Tifentale and Lev Manovich. Selfiecity: Exploring photography and self-fashioning in social media. In *Postdigital Aesthetics*. Springer, 2015.

[281] TIME. Instagram just hit the 500 million user mark. http://time.com/money/4376329/instagram-users/, 2016.

[282] TIME. Here's how much time snapchat users spend on the app. http://time.com/4272935/snapchat-users-usage-time-app-advertising/, 2016.

[283] Zeynep Tufekci. Big questions for social media big data: Representativeness, validity and other methodological pitfalls. In *ICWSM '14: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[284] José van Dijck. Chapter 4: Twitter and the paradox of following and trending. In *The Culture of Connectivity: A Critical History of Social Media*, pages 68–88. Oxford University Press, 2013.

[285] Diederik van Liere. How far does a tweet travel?: Information brokers in the Twitterverse. In *MSM '10*, pages 6:1–6:4, 2010. ISBN 978-1-4503-0229-6. doi: 10.1145/1835980.1835986. URL http://doi.acm.org/10.1145/1835980.1835986.

[286] Peter L.M. Vasterman. Media-hype: Self-reinforcing news waves, journalistic standards and the construction of social problems. *European Journal of Communication*, 20(4): 508–530, 2005.

[287] P. O. S. Vaz de Melo, L. Akoglu, C. Faloutsos, and A. A. F. Loureiro. Surprising patterns for the call duration distribution of mobile phone users. In *ECML-PKDD*, 2010.

[288] Pedro Olmo S. Vaz de Melo, Christos Faloutsos, Renato Assunção, and Antonio Loureiro. The self-feeding process: A unifying model for communication dynamics in the web. In *WWW*, pages 1319–1330, 2013.

[289] Maria Vegega, Brian Jones, Chris Monk, et al. Understanding the effects of distracted driving and developing strategies to reduce resulting deaths and injuries: a report to congress. Technical report, United States. Office of Impaired Driving and Occupant Protection, 2013.

[290] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[291] Di Wang, A. Al-Rubaie, J. Davies, and S.S. Clarke. Real time road traffic monitoring alert

based on incremental learning from tweets. In *EALS '14*, pages 50–57, 2014.

[292] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *WWW*, 2012.

[293] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *KDD*, 2011.

[294] Xin Wang, Tristan Gaugel, , and Matthias Keller. On spatial measures for geotagged social media contents. In *MUSE '14*, pages 35–50, 2014.

[295] Yang et al. Wang. A field trial of privacy nudges for facebook. In *CHI*, pages 2367–2376, 2014.

[296] Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2(2), 2014.

[297] Micah White. Clicktivism is ruining leftist activism. *The Guardian*, August 2010.

[298] Charlotte Wien and Christian Elmelund-Præstekær. An anatomy of media hypes: Developing a model for the dynamics and structure of intense media coverage of single issues. *European Journal of Communication*, 24(2):183–201, 2009.

[299] Shirley A. Williams, Melissa M. Terras, and Claire Warwick. What do people study when they study Twitter? Classifying Twitter related academic papers. *Journal of Documentation*, 69(3):384–410, 2013.

[300] Amy Willis. Mumbai bans selfies after 19 people die. http://metro.co.uk/2016/02/25/mumbai-orders-selfie-ban-after-19-people-die-5716731/, 2016.

[301] Ke Wu, Kun Zhang, Wei Fan, Andrea Edwards, and S Yu Philip. Rs-forest: A rapid density estimator for streaming anomaly detection. In *ICDM*, pages 600–609. IEEE, 2014.

[302] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *KDD*, 2010.

[303] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[304] L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.

[305] Songhua Xu, Hao Jiang, and Francis Chi-Moon Lau. Mining user dwell time for personalized web search re-ranking. In *IJCAI*. AAAI, 2011.

[306] Junting Ye and Leman Akoglu. Discovering opinion spammer groups by network footprints. In *COSN*, 2015.

[307] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *RecSys*, pages 113–120. ACM, 2014.

[308] P. Yin, P. Luo, W.-C. Lee, and M. Wang. Silence is also evidence: interpreting dwell time

for recommendation from psychological perspective. In *KDD*. ACM, 2013.

[309] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. *SIGCOMM*, 36(4):267–278, 2006.

[310] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: Discover spatio-temporal topics for Twitter users. In *KDD '13*, pages 605–613, 2013. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487576. URL http://doi.acm.org/10.1145/2487575.2487576.

[311] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2016.

[312] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[313] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[314] Kathryn Zickuhr. Location-base services. Technical Report Pew Internet and American Life Project, Pew Research Center, September 2013.