Fall 12-13-2021

Information retrieval from graphs with feature engineering

Dhara Shah

Information retrieval from graphs with feature engineering

by

Dhara P. Shah

Under the Direction of Dr. Robert W. Harrison

ABSTRACT

Information retrieval on graphs has applications in diverse areas, including analyzing social networks, computer security, and understanding the evolution of drug resistance in HIV. However, in each domain, there are relationships between data points that need to be preserved to extract meaningful information. The data sets in each area could be represented as data structures that could be embedded in graphs.

This dissertation seeks the approach of feature engineering for the information retrieval from two very different but very large datasets, namely genome sequencing of HIV protease to study the virus' resistance to the HIV drugs and text data of illicit groups from telegram, a social network platform. The goal of this dissertation is to demonstrate that the new, vital information such as understanding the evolution of HIV protease or predicting if the given message contains illicit information or not, doesn't have to be based on high-cost computing methods, and the resultant information is still comparable to its expensive alternatives.

To understand the evolution of HIV under the protease inhibitor drugs, this dissertation first demonstrates feature-engineering of the HIV protease, a complex protein structure, as a transitionally and rotationally invariant sparse vector representation preserving the relative positions of Amino Acids. This dissertation then demonstrates the effectiveness of this vector representation and understands the evolution of HIV protease as a minimal spanning tree. In the end, by understanding the branches of minimal spanning trees covering these vector representations of HIV protease through time, this dissertation concludes the important and new observations on the resistance of HIV.

To seek the classification of a message being illicit or licit, this dissertation first under-

stands the nature of the illicit texts in the financial fraud domain in terms of the special words used in known licit and illicit groups in different contexts and hence frequencies. This dissertation feature-engineers the texts of these groups as sparse vectors by constructing a new bag of words comprising these informative words. In the end, this dissertation demonstrates the effectiveness of these sparse vector representations by applying shallow classifiers determining the ownership of the message given two groups.

INDEX WORDS: Machine Learning, Graph Information Retrieval, Feature Engineering, Informed Bag Of Words, IBOW, Evolution off HIVPr, Telegram, SWED, RSWED, Minimal Spanning Tree

Information retrieval from graphs with feature engineering

by

Dhara P. Shah

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

December 2021

Information retrieval from graphs with feature engineering

by

Dhara P. Shah

Committee Chair: Robert W. Harrison

Committee: David Maimon

Irene Weber

Marie Ouellet

Yanqing Zhang

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

DEDICATION

I dedicate this dissertation to the two people who have have been the pillars of the most important opposite ends: My husband who kept me logical on the emotional side, and my adviser who kept me logical on the rational side. Thanks to your support, I could do this, and that too all over again, in a new domain and in a year and a half. Thank you for believing in me and help me believe in myself, and not letting me forget what, why and how.

This dissertation would not have been possible without Chirs' constant encouragement and guidance, Saba's support as a friend and a manager, Dr. Weber's encouragement and input.

It indeed took a village to make this happen! I attribute my sustenance through this time to my Mom, Dad, Gallu, Rono and Kayu, through their unconditional love and support, even when I had to start my dissertation from scratch after five years. This ride would have been rougher without Chinua, Karthik, Sameera, Tammie, Adrienne, Jamie and Paul. I thank Shawn, Andrew, Christina, Mehdi, and my doctors for helping me show up and stand literally and metaphorically.

ACKNOWLEDGEMENTS

- Dr. Robert Harrison for helping me to mathematically conceptualize the ideas of this research
- Dr. Irene Weber for enabling me with the knowledge of biology needed
- Dr. David Maimon and Dr. Marie Ouellet for guiding my understanding of the telegram from criminologists' eyes
- Dr. Chris Freas for introducing me and managing the right deployment, networking, and sustainability tools for the needs of this research
- Bhashithe Abeysinghe for helping me with NLP development and tools that resulted in a separate parallel research
- Fatemeh Aslanzadeh and Karthik Subramanian for help with the data mining and features parsing
- Sameera Tapuru for Data Analysis that resulted in a separate research
- Dr. Marco Valero for helping me sustain while working on the end part of this dissertation
- Dr. Yubao Wu for the input while exploring the initial concepts

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- APV: amprenavir
- IDV: indinavir
- LPV: lopinavir
- NFV: nelfinavir
- TRV: ritonavir
- SQV: saquinavir
- ATV: atazanavir
- TPV: tipranavir
- DRV: darunavir
- HIV: Human Immunodeficiancy Virus
- HIVpr: HIV protease
- SWED: Structure-Weighted Edit Distance
- RSWED: Radial Structure-Weighted Edit Distance
- KL: Kullback-Leibler
- SVM: Support Vector Machine
- ANN: (one layered) Artificial Neural Network
- MST: Minimum Spanning Tree
- RMSE: Root Mean Squared Error

- t-SNE: t-distributed stochastic neighbor embedding
- NLP: Natural Language Processing
- NLU: Natural Language Understanding
- BOW: Bag Of Words
- IBOW: Informed Bag Of Words
- HTML: HyperText Markup Language
- TF-IDF: term frequency–inverse document frequency
- logreg: Logistic Regression

PART 1

INTRODUCTION

Information retrieval on graphs has applications in diverse areas including the analysis of social networks, computer security, and understanding the evolution of drug resistance in HIV. However, in each of these areas, there are relationships between data points that need to be preserved in order to extract meaningful information. Hence data-sets in each area could be represented as data structures that could be embedded in graphs.

Social networks are inherent part of billions of people today. Users are the primary consumers of the social network platforms, where they can communicate through messages with other users. With huge fraction of human population having social persona through a variety of social networks, understanding human behavior in various aspects is equivalent to analyzing social network data in different ways.

Social networks also allow the person to create a cyber persona that couldn't be linked to the user in the real world directly. A user could have several different cyber personas on a single or multiple social networks. In addition, cyber social networks also provide world wide platforms that are spanned through several national and international jurisdictions. This means that social network platforms could also be used to organize criminal activities. With an avalanche of information, its really hard to extract criminal information from the social networks and identify criminal activities of a single person or create a criminal profile of people in certain criminal domains.

The most popular social networks such as Facebook, Twitter, and Whatsapp are heavily protected against criminal activity as well as legal but harmful idiosyncrasies. Therefore many people have turned to Telegram, a social network known for providing anonymity to users and lack of scrutiny of the hosted contents. Hence understanding telegram graph through various information retrieval tools means understanding the state of the art of

criminal domains.

Section 3 describes my work on telegram's financial fraud domain. The introduction to which is provided in 1.1.

Section 2 describes my work in using machine learning to understand evolution of drug resistance in HIV, introduction to which is provided in 1.2

HIV, a highly mutative virus has exponential mutation rate because of is error prone replication process. Due to this exponential mutation rate, the number of HIV virus mutations found is very high; several hundred mutants of the virus are found in a single patient. This means that the drugs to treat this virus will have to target the replication steps that are unique to the virus.

One of these virus specific mutation steps involves producing HIV protease, a protein that the virus uses in the process of maturation after budding. Maturation is a critical step in the viral life cycle and viruses that do not mature cannot infect cells. There is a class of HIV curbing drugs that inhibits the protease and thus prevent the virus from becoming infective. However, the several mutants of HIV virus in the infected person's body respond to the HIV drugs with different sensitivity. A drug resistance HIV will still mature and be infective even when exposed to levels of the inhibitor that would effectively prevent maturation in wild type virus. Through our research presented in this dissertation, we track the HIV protease as a graph. The nodes of this graph are the representation of the protease and edges of this graph represent the molecular similarity between the distinct protease. This design of the graph is formed under the assumption that similar molecules have similar properties.

1.1 Feature Engineering and Machine learning on telegram's financial fraud echo-system

1.1.1 telegram as social network

Telegram is a cyber-social network that is based on the basis of 'freedom of speech'. Telegram appeals to the users because of its fast end-to-end delivery, the type and vol-

ume of multimedia being communicated and stored without scrutiny. Telegram is also a popular medium because of its no-censorship policy on its content, while protecting the user's anonymity with end-to-end encryption. Telegram also provides highly tunable privacy options that no other platform provides, including secret chats, connection proxies, timed auto-deletion of messages and no-trace-left exit from chat rooms.

1.1.2 entities in telegram

A telegram entity is a unit that is recognized distinctly as a special function/privilege class in the telegram back-end. Telegram is comprised of three kinds of entities – messages, entities which can read or post the messages, and messaging boards.

Messages are the entities that form the communication in telegram. A message could hold text, multimedia, executable files, GIFs, emojis, etc but its wrapped into a general class of message telegram, so on the higher level, hence would be identified as a 'message', for the purpose of this dissertation.

The entities that can read and write the messages are user accounts and bots. A user account is defined by a unique phone number. A bot is a user with automation privileges, and is recognized distinctly from the normal user in the telegram back-end. A bot is always associated with a user account, but a user account can have several bots associated with it. For the purpose of this dissertation, these kinds of entities would be referred as users unless otherwise specified.

Messaging facilities could be sub categorized as groups, channels and chats. Groups are the message boards where multiple users can read and write messages. Channels are broadcast only bulletins where only the privileged users can post messages, but everyone can read them. Chats are personalized interactions between two people. Each of these messaging facilities is highly customizable, and telegram back-end identifies each of them separately, so essentially there are several distinguishable sub-types of these, but for the purpose of this dissertation, we only consider the public groups where everyone can post and read the posted messages.

1.1.3 telegram for illicit activities:Ease, Anonymity and no scrutiny

Telegram's patronage is users from several countries – this is evident from the huge amount of groups formed in different languages and vast participation from the patronage of these groups. These users have formed groups and channels on all possible topics, most of them consist of non-criminal content. However, due to telegram's policy of not scrutinizing the message content and protecting user's identity by end-to-end encryption, telegram is also the primary platform for several illicit activities.

The headquarters and the infrastructure of telegram are hosted around the world and are kept secret to avoid the international criminal laws. Without the scrutiny from telegram, these illicit activities also have a vast range; protests/ hate/ terrorist/ extremist groups, illegal mutilation and sales of animals or animal parts, selling popular game/ movie/ travel tickets at inflated prices, drugs/ weapon/ prescription sales, financial fraud etc [1]. For the purpose of this dissertation, we only focus on illicit activities pertaining to financial fraud.

For purpose of this dissertation, we define the domain of illicit financial fraud as activities pertaining to intent of illicit money transaction. Examples of these activities are including, but not limited to money transfer between accounts/ countries for the sake of tax fraud or hiding transactions from a government, stolen credit card information for unauthorized use, bank check fraud, identity theft for the purpose of financial gain, free online accounts and gift-cards bought from unauthorized use of money, tutorials on technicalities and social engineering aspects of these activities. These illicit activities occurs as messages pertaining to these activities posted in groups, channels, and one on one chats, where the users send or receive these messages.

Setting an account on telegram is free, anonymous, requires zero investment or technical skills. The only pre-requisite to conduct an illicit activity on telegram is to get a phone number that is capable of receiving one text message for the sake of verification on telegram, and once set, it could be proxied to the sole usage of primary telegram account. With the revolution of telecommunication protocols such as VoIP, the phone number used to setup a telegram account could be a virtual service that could be free on the web interface, without

any strings attached to the user's real identity.

1.1.4 Telegram as a backup to dark-web and scrutinized social media

The ease of having an anonymous telegram account also makes telegram an excellent backup for illicit activities on dark-web, as well as criminal activities on more popular but heavily scrutinized online platforms such as Facebook, Instagram, Whatsapp, Twitter, Linkedin, Tiktok, Signal, Discord, and Snapchat.

1.1.5 Observing financial fraud ecosystem on telegram

In order to understand the financial fraud ecosystem within telegram, we have registered as users with multiple phone numbers and joined several groups and channels pertaining to financial fraud. In particular, we joined and monitored 100+ public groups for a year to hypothesize the behaviors of the entities involved. Our role there is that of a spectator, just observing the day-to-day business of financial fraud and not engaging in the telegram ecosystem. Occasionally, we have also engaged in one on one interactions with some users that have reached out to us posing as sellers. We have used these opportunities to verify the knowledge we already had, and enhance it further, and hence strengthening our hypotheses of the types of users and groups, their authority and influence.

The absence of peer reviewed qualitative or quantitative studies on the financial fraud domain in telegram has left us to hypothesise our own empirical observations and test them. Henceforth going forward, we are only able to explain what we have observed without being able to cite similar peer reviewed observations. Our observations could be confirmed by anyone having a telegram account and observing any random subset of groups of financial fraud domain even by a quick glance.

We have empirically observed the following in the financial fraud ecosystem of telegram over the period of last two years:

1.1.6 Types of groups

Groups could be set to be either private or public. Private groups are the groups where the membership is invite only, invitation being a link that has an expiration time. The groups are created by users that serve as admins of the group. The public groups are the groups that can be discovered by their name and anyone can join it.

Each financial fraud group has a life-cycle with the phases of the creation of the group, the peek and the absence of the activity in that group. Once created, users can join the group. Most users join a group directly by searching for the group handle in the search bar in the telegram GUI and seek membership. In the financial fraud ecosystem, users also commonly join a group indirectly when added to the group by another user. The users in this group then start exchanging messages, where the messaging activity peaks eventually. The end of a group's life-cycle is reached abruptly or organically. Sometimes, the administrator's decision to shut down the group and move it to a different name, or abrupt abandonment ends the group's active participation abruptly. A group reaches to a slower and organic end when active people leave the group for various reasons, and there is no activity for increasingly long periods. for the purpose of this study, we consider the group inactive when it seizes to be an active part of the financial fraud ecosystem either by removal or by non-activity, even if the group still exists as a telegram entity.

we have observed that the private groups have a very short life-cycle and become monotonous for the discussion content almost immediately after the formation. Also, the channels are essentially used as bulletin boards, and the effective channels are usually run by the influential administrators of public groups. Hence, for the purpose of this study, we only chose to observe the public groups that have more than 50 users, and are active almost every day for last two months on the day we mined the groups for their data.

1.1.7 Identifying active public groups

By observing this life-cycle of the financial fraud groups, we could deduce that the groups that are public, have more than 50 users, and are active almost every day for last two

months, could be considered active part of the financial fraud ecosystem. For the sake of this study, we only focus on these groups that lead to quantitative studies for the sustainable and substantial user to user interaction.

1.1.8 Types of users

From a user's perspective, we have identified several roles of the participating users in the financial fraud ecosystem. Just like any business, primarily, this ecosystem sustains on buyers. Buyers are the users who pay money for buying illicit goods/ knowledge/ services. Users who provide to these buyers are sellers. Not all sellers sell the promised or effective products. The sellers who scam the buyers are often referred to as scammers. With the lack of regulating laws, there are some third party users that work as middle-man, where they guarantee the buyer that the desired goods will be delivered for the money. These users are referred to as escrow. We have found that the reputed escrows are praised often, and buyers vouch for them, but the validation or reason of their effectiveness is still under investigation by us. There are also some users who advertise a particular buyer or group. Some users also engage in one on one conversation with other potential buyers. Often times, these users also add other users to groups pertaining to same or similar kind of fraud. These users are referred as promoters. The motives of promoters vary depending on their role in the business. Some promoters are administrators of certain groups, and are trying to popularize it by praising or promoting it on other groups. We also classify the users who add other users to a group as promoters. Most of these promoters gain some monetary benefits from the promotion, but there are some users who promote certain groups or escrows just to gain favor with the influential users or a promotion in their role. Sometimes the groups stipulate posting privileges only if a user promotes the groups with either certain number message forwards or by adding a certain number of users in the group. For the purpose of this study, we have avoided these groups to be included in our data-set.

1.1.9 Chatbots as users

Not all users in this ecosystem are human. Chat-bots, the users that are enabled with programmable automation capacities, are also integral part of this ecosystem. Role of these chat-bots varies depending on their programmed tasks. We have observed the chat-bots as helpers to admins in a group, and hence in the capacity of promoters and admins. We have observed chat-bots posting event based messages. For example, when a user joins a group or posts messages with certain content, add or remove users, or walk a user through certain steps to complete a process etc. The creator of a chat-bot is a user who can dynamically change the functionality of a chat-bot. On occasions, we have also seen chat-bots chatting with users, but this could be either a functionality of the underlying program or the creator of chat-bot chatting through it.

1.1.10 Time-frame and entities across

As described in 1.1.5, the topology of the financial fraud ecosystem is highly dynamic. In particular, the groups have life-cycle, the users and groups can change or delete their user-name or group-name without consequences, and the messages could be deleted on schedule without leaving a trace.

Also, telegram as a platform removes all the digital footprints that could lead to real identity of a person. In particular, telegram removes the timezone information by converting the timestamp of a message to UTC time. Telegram also removes the metadata of any files that are uploaded to the platform, hence anonymizing the location, name, or other information that might lead to the real identity of the user.

Telegram identifies each user with a user id, that is uniquely associated with the phone number that was used to create the user-account. User has option to hide their phone number from public. Similarly, each group is also given a unique group-id at the time of its creation that could not be changed.

The groups and users are searchable in the telegram GUI by their telegram handle, the unique string that could uniquely identify a user as a username. For the purpose of this

dissertation, we will use telegram handle and username interchangeably. Unlike user/group id, the users or groups are not required to have a username, and this username could be hidden, removed or changed any time.

Also, a user or a group is allowed to have a title. This title could hold the group's name or a user's name. This title is the header that is seen by public when a group or user is viewed. This title is different from the username of the group or group/user id and is independently managed by a user or an admin. The title string doesn't need to be unique like username, and hence multiple users can have same names displayed and multiple groups could have same titles. Also, this title could be changed frequently without leaving the trace of the change.

For the purpose of this study, we only identify groups, messages and users by their telegram ids. Going forward, the reference to distinct groups/messages/users means the groups/users/messages having distinct ids.

The focus of this study is to understand and quantify typical behavior of financial fraud domain. We did not wish to identify or quantify the anomalies quite yet. By observing about 100 financial fraud groups for a year, we determined that an observation window of about 60 consecutive days covers all the typical event patterns of the financial fraud ecosystem. The users, groups and messages could be deleted without trace. However, absence of certain messages or users would not contribute towards statistical variance of the common patterns.

1.1.11 Active efforts of identity obfuscation: Groups and Users

We have observed that users and groups actively try to misidentify or obfuscate their identity, as in, some users try to match their username and profile picture to someone influential, as well as a same user could deactivate his account and come back as a user with new user-id as a "new" user resuming his criminal activity. We do not know the reasons behind this obfuscation, but our empirical observations suggest that misidentifying themselves as an influential user attracts criminal buyers. Also, deactivating their account severs ties to the dissatisfied buyers and bad reputation to the account, while its hard for

people to realize that the same user came back with a different account.

The same phenomenon occurs on the group level as well. Groups who have handles and group profiles similar to that of the popular groups attract more subscribers. Also, re-packaging an existing but dead group as a new group and adding the same members to it allows the group to attract new and active criminal members while pretending to have been popular already.

Due to these acts of obfuscation, its really hard to keep track of the same semantic entity across the time.

Types of messages in a group The main media of the communication between various users is messages. Messages are the short strings of text that the users post on groups to communicate with each other. We have observed that in the criminal ecosystem of telegram, the public groups contain mainly three kinds of messages: The advertisements of the illegal goods posted by the sellers, the short messages that express interest in buying the illegal goods, and the conversations between the users. These conversations could be one sided messages such as welcome messages / removal messages from the automated chat-bots or they could be two sided conversations between users.

1.1.12 Telegram study: our goals

The main goal of this study is to understand the relationship between the entities involved in the financial fraud ecosystem. In particular, could we identify the ownership of the message? As in, if we know enough of the examples of messages exchanged financial fraud ecosystem, then given an unseen message,Could we classify if

1. the message is pertaining to financial fraud?
2. the message is a conversation or an advertisement of illicit activity?
3. the message is written by a chat-bot or not?

In particular, we seek to lay the ground work for answering questions like:

- Can we take a fair sample of the financial fraud ecosystem, and study the entities involved for their role in the ecosystem?
- Can we tell the difference between and the importance of the topics discussed in the groups by comparing their texts to each other?
- for the entities related to illicit activities, can we quantify some of the qualities based on the empirical evidence, and provide a generalization of the rule in the ecosystem?
- Can we profile the users, groups and messages?

by developing the machine learning, data analysis, and natural language tools required to study social networks like Telegram. The answers towards these questions has been investigated in another MS dissertations by Sameera Tupuru and Fatemeh Aslanzadeh working with me in this area.

1.1.13 Natural Language Approaches

The language in the telegram criminal network is English-like, but like vernacular English. The drawbacks of telegram language are that it doesn't follow the word distribution of the vernacular English, is jargon heavy, doesn't always follow the grammar structure of English, uses non-English words with English text. For the words that pass as English, the context of these words is very different and hence the meaning of these English words is different than that of vernacular English. Due to these limitations of the language, the conventional NLP/NLU utilities do not come to our aid.

Given two groups/users, the goal of this research is to find the likely owner of the given message. Our empirical observation pointed that there were some words that were used in different context in the two given sets of messages. This implied that their frequencies were different in both the sets – one was significantly higher than the other. We also looked at the words that were used in highly different frequencies, and that too gave us a clue that a message belonged to one set or the other.

We converted this observation into a distinguishing feature by the following method: We first found all the common words from the two given, labeled sets of words. We then calculated relative Kullback measure of these word. For example, a word that appears in both sets of groups has frequencies p_1 and q_i in both the sets. Then, the relative Kullback measure of this word is $p_i log(p_i/q_i) - q_i log(q_i/p_i)$. For all the common words in the given two sets, this relative Kullback measure forms a two tailed distribution. The words at the end of each tail are the idiomatic features that are so unique that determine the ownership of a new message.

In the light of this words, a new message can now ignore all other words but these words, and hence the text is a sparse representation of these informed bag of words. Using this informed bag of words, now we can form all the massages as feature vectors with one hot encoding and build models for the supervised learning.

1.2 Feature Engineering and Machine learning in HIV Protease Protein Structure

AIDS (Acquired Immunodeficiency Syndrome) is a disease caused by highly mutative virus, categorized as HIV (Human Immunodeficiency Virus). This virus uses HIV-protease (PR) to build virus DNA with host cell proteins. PR blocking antiretroviral drugs are shown to be effective therapy to the HIV infection. These drugs have different kinds of HIV-protease inhibitors, preventing HIV-protease to be made in the cell. Eight of the most popular inhibitors are amprenavir (APV), indinavir (IDV), lopinavir (LPV), nelfinavir (NFV), ritonavir (RTV), saquinavir (SQV), atazanavir (ATV), tipranavir (TPV) and darunavir (DRV).

HIV still mutates to resistant strains under these drugs making the drugs ineffective. This could be prevented by predicting the drug resistance. In particular, the mechanism to predict which HIV-protease inhibitors stop to be effective (are 'resistant') to the production of the HIV-protease could mark the underlying virus as a infectious mutant.

Biological structures, such as a proteins, are made of fixed set of molecules called Amino Acids. These Amino Acids are situated at specific relative positions with each-other. To

represent a protein mathematically, the absolute positions of the proteins is not important, since they could exist anywhere in the living organism. For example, the same proteins exist in each cell of human body, and their position in the cell is dynamic. Thus, To represent a protein, the mathematical measure should be rotationally and translationally invariant. The proteins are uniquely defined by their Amino Acids' relative positions to each-other. This set of uniquely relative positions fits the bill of being rotationally and translationally invariant, and thus could be used as a vector representation of the protein.

Proteins are central part of viral infection, where virus uses special kinds of proteins known as protease to dissolve the host cell walls to penetrate the cell. Drugs preventing viral infection often inhibit the protease of this virus to be ineffective to stop the infection. Being a protein, a protease could also be represented as a spatial structure. With this as a feature vector, a protease could be tracked to see if the virus is resistant to the drug. A popular method to identify these protease as a spatial vector is to study its crystal structure. for example, the protease of T4 virus – T4-lysozime [2], [3] has been studies in details with different vector represents of its crystal structure. This method of representation yields to dense representations of proteins.

1.2.1 Protein representation through Voronoi Diagram

Given a set of points in a metric space, a Voronoi diagram gives a unique, tessellated bounding boxes of these points. Thus, Voronoi diagram preserves the information of the relative locations of the underlying points. Hence, when a set of points is presented as a unit where only the relative positions of these points is important, they could be represented uniquely by their Voronoi tessellation. Conversely, given a Voronoi tessellation, the underlying points could be unambiguously created in the same relative positions. Once created for a specific protein, this unique Voronoi tessellation could be used to identify that specific protein, irrespective of where it exists.

Giving a formal definition of Voronoi Diagram, let $P = x_1, \cdots x_m$ be a set of points in $\mathbb{R}^n$, where $m \geq 2$ and $n \in \mathbb{N}$. $V(P)$, the Voronoi diagram/tessellation of P is defined as

$V(P) = \{V(x_1), \cdots V(x_m)\}$ where $V(x_i) = \{x \in \mathbb{R}^n | \parallel x - x_i \parallel \leq \parallel x - x_j \parallel \forall j, j \neq i\}$.

Small position change in a point only causes a few boundaries of this diagram to change, the majority of the diagram remains unchanged.

A Voronoi diagram, as a data structure, is made of an arbitrary number of boundaries that might be infinite. Thus, in terms of computer science, normalizing a Voronoi diagram as a homogeneous data structure is NP hard.

1.2.2 Delaunay Triangulation:Dual of Voronoi Diagram

A graph is a set of nodes and edges. A dual graph of the given graph is obtained by representing each face of this graph as a vertex and connecting each pair of faces through an edge.

If each boundary of the Voronoi diagram is considered an edge, and the point where two or more edges meet, is considered as a node, then the dual graph of this graph is a Delaunay triangulation.

Given a set of points in a Euclidean space, each set of three distinct points has a unique sphere passing through them. Through all possible combinations of three distinct points, if the unique sphere passing through them contains any other point of the given set in it, discard that triplet. Hence, at the end, only the triangles comprised of the three points whose unique sphere passing through them does not contain any other point from that set. This is the Delaunay triangulation of the given set. Each set of points in a Euclidean space has a unique Delaunay triangulation.

Being a dual of a Voronoi diagram, Delaunay triangulation is also capable of representing a biological structure. Delaunay triangulation represents a biological structure as sparse adjacency list of the nodes (molecules). In comparison to Voronoi node-edge representation of the bounding boxes, the Delaunay triangulation has less nodes and edges by magnitudes. Also, Delaunay triangulation is a robust geometric structure, as in small changes in the positions of the nodes do not change the Delaunay adjacency list. This is hence a robust representation for the biological structures since the position of molecules

in biological structures maintain a range of distances, but it doesn't change the Delaunay adjacency list.

1.2.3 Protein representation as Delaunay triangulation

A protein is a 3D chain of several Amino acids. Each Amino acid could be located by the alpha-carbon at its center. Each protein could be represented as unique Delaunay triangulation using their alpha-carbon positions. Once the triangulation is created, the Delaunay adjacency list could be generated for all the Amino acids involved in that protein. There are 20 types of Amino acids. Hence, each edge of a protein's Delaunay triangulation could be classified as an edge between Acid of type i and Acid of type j, where $1 \leq i, j \leq 20$. Hence, a $20x20$ edge metrics of all the possible types of Delaunay edges among the 20 types of Amino Acids could be constructed. where the cell (i, j) represents the number of edges in the Delaunay triangulation between the Amino Acids of type i and j. The Delaunay triangulation being an underacted graph, this edge-count metric would be a symmetric matrix. The unique information of this matrix lies in its upper triangle and diagonal - 210 cells out of the 400 cells.

PART 2

EVOLUTION OF DRUG RESISTANCE IN HIV PROTEASE

Drug resistance is a critical problem limiting effective antiviral therapy for HIV/AIDS. Computational techniques for predicting drug resistance profiles from genomic data can accelerate the appropriate choice of therapy. These techniques can also be used to identify protease mutants for experimental studies of resistance and thereby assist in the development of next-generation therapies. Few studies, however, have assessed the evolution of resistance from genotype-phenotype data.

The machine learning produced highly accurate and robust classification of resistance to HIV protease inhibitors. Genotype data were mapped to the enzyme structure and encoded using Delaunay triangulation. Estimates of evolutionary relationships, based on this encoding, and using Minimum Spanning Trees, showed clusters of mutations that closely resemble the wild type. These clusters appear to evolve uniquely to more resistant phenotypes.

Using the triangulation metric and spanning trees results in paths that are consistent with evolutionary theory.The majority of the paths show bifurcation, namely they switch once from non-resistant to resistant or from resistant to non-resistant.Paths that lose resistance almost uniformly have far lower levels of resistance than those which either gain resistance or are stable. This strongly suggests that selection for stability in the face of a rapid rate of mutation is as important as selection for resistance in retroviral systems.

2.0.1 Background

Selection pressure due to the widespread use of anti-retroviral therapy [4] makes Human Immunodeficiency Virus (HIV) a valuable model for studying evolution. HIV/AIDS is a major pandemic disease[5] where more than 37 million people have been infected. Currently, about 60 percent of the infected people receive anti-retroviral therapy. Antiviral drugs block viral replication by targeting the viral enzymes, protease, reverse transcriptase and

integrase, HIV entry and fusion to the host cell. Further progress in treating the disease is hampered by the selection of drug-resistant viral strains. Since the conversion from the RNA genome to DNA is error-prone, HIV mutates rapidly[6]. HIV readily forms quasi-species and distinct viral strains. Thus, HIV possesses the prerequisite high degree of variation for the rapid evolution of drug resistance. In addition to studying drug resistance to enhance the development of novel therapeutics, studying the evolution of drug resistance can help define the optimal strategy to overcome drug resistance with current approaches.

HIV protease is an excellent model system due to its relatively small size and the extensive data for sequence variants and structures [7]. The protease acts as a dimer of two 99-residue subunits. Experimental studies [8] and theoretical analysis [9] of the protease mutants suggest that many of the secondary mutations contribute to the survival of the original resistance mutations by improving the effectiveness of the protease for viral replication. These findings suggest that initial mutations introduce resistance and further selection improves the fitness of the enzyme. Therefore, we expect to see linked sets of mutations in the resistance data, and have developed an analysis based on Minimum Spanning Trees to detect and analyze these linkages.

Previous studies by our group and others show that machine learning can accurately predict resistance phenotype from genotype data for HIV protease and reverse transcriptase [10],[11],[12],[13], [14],[15], [16], [17]. We have found that including structural data with the sequence using Delaunay triangulation is an especially effective representation for machine learning [18]. The combined sequence and structure information is compressed into a single 210-dimension vector for each mutant. In essence, this approach is an sequence edit distance weighted by the most significant local contacts in the protein. It is nearly a linear metric space [14]. Simple machine learning approaches such as a linear SVM and k-nearest neighbors are able to reliably classify resistance data with this encoding of sequence and structure. This approach is a marked contrast from other work where complicated or deep machine learning approaches are used [19],[17],[16]. It creates the ability to use the features for more than simple classification or regression. Our previous work [12],[11],[10],[14],[20],[18] has

concentrated on developing models for predicting the resistance to single inhibitors. Shen et al [14] and Pawar et al [20] demonstrated classification accuracies higher than 99%. However, many of the resistant strains have lost susceptibility to all clinical inhibitors. It is important, therefore, to apply machine learning to the prediction of resistance to multiple inhibitors. Our previous work [20] showed that there was significant cross-prediction accuracy where models trained on one inhibitor predict the response to other inhibitors. This suggests that there are commonalities in resistance mechanisms and the first step to studying these commonalities is to build a machine learning model that describes them. This model can then be used to select sequences for expression, characterization, and structural analysis.

Gene trees are a major tool in the construction of molecular phylogeny [21], [22], [23], [24] and they have been applied to HIV [25]. Much of the existing work has been applied to estimate gene flow between species, gene duplication, and horizontal transfer. Typically, sequence distances are used to estimate similarity between genes and then a graph is constructed that reflects the relationships between the genes. The graph is a tree in the absence of horizontal transfer and gene duplication. There are subtle but important differences between the standard use of gene trees and this study because mutational data in the HIV protease gene do not involve gene flow between species, gene duplication or horizontal transfer. This paper examines the onset of speciation or quasi-speciation under the selection pressure of clinical treatment with potent protease inhibitors. It combines our highly effective representation of structure and sequence with well-understood algorithms for building minimum spanning trees (MSTs) to estimate the evolutionary properties of HIV response to drugs. Since this measure is linear or nearly linear and possesses metric properties, it should be an effective proxy for evolutionary distance. MSTs will serve as a first approximation to the gene tree.

The development of "super-resistance" is a related question. Naive selection theory suggests the "first past the post" mutations, those that are sufficiently resistant to allow HIV to reproduce in the presence of inhibitors, will be a majority of those selected. If drug resistance alone is sufficient for evolutionary selection, why should new mutations accumulate

in the protease? Yet there are many examples of highly resistant proteases bearing different sets of multiple mutations which are believed to enhance viral replication [26]. The pattern of resistance acquisition and loss along branches of the MSTs sheds light on the selective pressures for drug resistance. The virus must not just become resistant, but must retain resistance and effective replication in the presence of a high mutation rate.

2.0.2 Genotype-phenotype data from the Stanford HIVdb

The collated data in the Stanford database [27] is a valuable resource for computational analyses. The data consist of the sequences of HIV drug targets, including HIV protease, and resistance measures. The database is curated and updated regularly to reflect the current status of drug resistance in HIV. We used the filtered phenosense data for this paper [28].

2.1 Results

2.1.1 Resistance Classification and Regression

The linear SVM was used to classify if the HIV protease sequence is resistant or not based on the threshold of 3.0 as defined in the Stanford database (shown in Table 1). The data are well-balanced for all inhibitors with the exception of Darunavir. Both the SWED and RSWED were used to train two different models with 3-fold cross validation. The quality of the prediction shows that our data were successfully updated. Table 2 2.2 shows the classification accuracy for pairs of inhibitors. Note that while there is some correlation between different inhibitors, there are significant differences between them. Table 2.3 shows the results for triples of inhibitors. Only a subset (those with ATV) is shown to conserve space, but the results are similar for all triples with both the RSWED and SWED metrics.

In addition to single inhibitors, classification of the resistance for all pairs and triples of inhibitors was done. In all cases, a high classification accuracy ($> 98\%$) was seen. Therefore, it was important to examine regression, where the magnitude of the observed effect is predicted. This is a more difficult measure than binary classification. Regression was preformed using random forest regression.

Figure **??** shows the RMSE as a function of the training fraction for cross-validation. A training fraction of 0.66 corresponds to 3-fold cross validation (2:1 ratio) and 0.2 is an inverted 5-fold cross validation (1:4). Since the range of observed values for the data is between 0 and 100, an RMSE < 0.1 corresponds to a high degree of accuracy. Figure 2.2 shows the distribution of RMSE for regressions over each pair and triple of inhibitors. The correlation coefficients were in the high range from 98% to 99%.

2.1.2 Spanning Trees

Figure 2.4 represents the spanning trees of a random 10% split of data with ATV. The nodes in these graphs represent the vectors generated by the upper triangular matrix with average distance and count, respectively. These spanning trees are calculated with respect to l_2 distances when the nodes are represented by distance and count vectors, respectively. The nodes that are resistant with value bigger than 3 for inhibitor are represented as green, and the non-resistant nodes are represented as red. Empirically, the spanning trees for all splits with respect to all the inhibitors have similar visualizations. The centers of these trees are the nodes whose sequences differ at most in two places from the standard wild type HIV-1 protease sequence of the group B sub-type M. Consistent with the high mutational rate of HIV, both resistant and susceptible strains develop differences from the standard sequence in a similar manner.

2.1.3 Path statistics in the spanning trees

Since the paths or branches in figure 2.4 appear to show the selection for resistance early in mutational history, followed by its conservation over time, it is necessary to examine the behavior of resistance along the branches of the tree. The sequences at the roots of the tree are close to the reference sequence and the branches, both resistant and non-resistant, show increasing numbers of mutations as they move from away from the center.

The paths fall into five general categories, those that: remain below the resistance threshold, gain resistance, lose resistance, remain above resistance threshold, or cross the

threshold multiple times, creating a spiking pattern.

Due to the density of the data, we summarized the gain, loss and spike patterns by plotting the mean value of resistance against the fraction of the path which is above the resistance threshold. Figures 2.72.9,2.11 represent the scatter plots of paths that gain, lose or spike in resistance. Each dot in these figures corresponds to an individual path. Figure 2.12 shows the histogram of the variance of paths above the resistance threshold. Most of the paths that are resistant have low variance,s which indicates that the magnitude of the resistance is stable, and therefore there is selection for stable resistance in the presence of high mutational rates.

2.2 Discussion

This paper demonstrates three points. First, it shows that SWED and RSWED measures still work well for classification and regression of resistance. This result is important since the sequence-structure representation was recalculated when the database was updated. Second, it shows that these representations, when used to generate an MST, appear to be valid proxies for evolutionary or mutational distances. Finally, the trajectory of resistance along individual branches of the trees suggests that the selection pressures for resistance are more complicated than would be naively thought.

2.2.1 Classification and Regression

With an elegant encoding, in this case the SWED and RSWED, even simple shallow learning algorithms like the SVM can achieve high accuracy. The accuracy in this paper is better than we achieved earlier, and we hypothesize that this is due to using better and more complete data. Including features of the geometry (amino acid positions), along with the labels (the sequence), results in an encoding for a physical object that captures most of the essential information.

2.2.2 A Proxy for Evolutionary Distance

Defining the evolutionary distance between two individual genomes is an open problem. Obviously, the distance must reflect mutations, but in a highly mutable system like HIV, straightforward counts of mutations can be misleading because the probability of a reverting mutation is relatively high. Therefore, including structural or biochemical information to assess the importance of individual mutations should improve accuracy. The SWED and RSWED measures include structural information. Figure 2.4 shows a visualization of the MST derived with both measures. The sequences at the roots of the tree are close to the reference sequence, while the branches, both resistant and non-resistant, show increasing numbers of mutations as they move away from the center. Interestingly, many of the branches maintain resistance or non-resistance during evolution. Quite often an initial single or double mutation becomes resistant and the resistance evolves further with additional mutations.

2.2.3 Behavior of the Branches

Analysis of the branches shows several interesting results. Most importantly, it shows that selective pressure for resistance is complicated. Figures 2.7, 2.9 and 2.11 show the relationship between path length and resistance for both paths that gain resistance and those that lose it. The naive model of selection would expect that viruses would evolve to be just resistant enough to replicate in the presence of inhibitors. Resistance along a branch or path shows significant differences from this naive model. Paths that maintain resistance tend to increase resistance to high levels. However, some paths may demonstrate "spiking" where they become highly resistant and then approach lower resistance levels. Paths that lose resistance inevitably are never highly resistant. This result strongly suggests that there is an additional selective pressure to become highly resistant. In the presence of high mutation rates, molecules that are "just resistant enough" will readily lose resistance. Proteases that are highly resistant could require many mutations to lose resistance.

It is clear in figures 2.7, 2.9 and 2.11 that there is some structure to the relationship between resistance and path length. The structure could reflect paths that have the same

root and diverge at some time during viral evolution. As a first pass at analyzing this relation ship, we clustered paths using the dbscan [29] algorithm as implemented in python scikit-learn [30] library. The similarity of paths starting from the same root is shown for a representative sample in 2.5. That these points appear to lie on smooth curves suggests that the structure seen in 2.7, 2.9 and 2.11 is due to paths that diverge during evolution.

2.3 Conclusion

A simple measure that combines structure and sequence is highly effective for classification and regression of drug resistance in HIV protease. Unlike pure sequence features, shallow learning, even simple shallow learning algorithms like the linear SVM, produce accurate results with this representation. In addition to clustering and selecting sequences for experimental study, the measure can be used for calculations the probe the evolutionary relationship between isolates of HIV. Our results suggest two major points for evolution of resistance. First, there is a conservation of resistance. Isolates become resistant early on and then tend to stay resistant. Second, there is a selective pressure for isolates to become highly resistant over time. Isolates that do not become highly resistant tend to lose resistance. This suggests that robustness with respect to mutation and change is an important selection pressure in evolution.

2.4 Methods

The methods in this paper range from preparing the data (Data Expansion and Vector Generation) to machine learning (Classification and Regression) and the development of models of evolution. A new expansion of the data was needed as the Stanford database was updated from the version used in previous work. This was a major update where the curators cleaned up the data. Because new data were generated it was necessary to show that our methods still worked. Classification and regression showed that the machine learning approaches are still highly effective. MSTs and analysis of the branches or paths in the trees was performed to inform hypotheses about selection due to drug pressure in

Inhibitor	fraction resistant	fraction susceptible	accuracy	F-score
FPV	36.4	63.6	99.5	99.5
ATV	21.5	78.5	99.7	99.8
IDV	33.8	66.2	99.6	99.7
LPV	34.5	65.5	99.6	99.6
NFV	27.3	72.7	99.6	99.7
SQV	68.3	31.7	99.6	99.6
TPV	60.4	39.6	99.7	99.8
DRV	97.1	2.9	99.92	99.93

Table 2.1: Classification statistics for HIVpr. Fraction of resistant vs non-resistant inhibitors for all inhibitors. Other than for Darunavir, the datasets are well-balanced. The 3-fold cross-validated accuracy and F-scores are shown for the count vectors using a linear SVM.

Inhibitor	Inhibitor	Pearson R	Accuracy
ATV	DRV	0.6444	99.8577
ATV	IDV	0.6139	99.7566
ATV	LPV	0.3535	99.8217
ATV	NFV	0.8655	99.808
ATV	SQV	0.9175	99.796
ATV	TPV	0.3837	99.808
FPV	ATV	0.617	99.8168
FPV	DRV	0.8092	99.8715
FPV	IDV	0.8215	99.8682
FPV	LPV	0.8694	99.8907
FPV	NFV	0.7385	99.8156
FPV	SQV	0.4485	99.8219
FPV	TPV	0.4229	99.894
IDV	DRV	0.4766	99.6974
IDV	LPV	0.8671	99.7512
IDV	NFV	0.8189	99.6915
IDV	SQV	0.4657	99.6959
IDV	TPV	0.4373	99.8133
LPV	DRV	0.3024	99.8013
LPV	NFV	0.6433	99.7536
LPV	SQV	0.1664	99.7166
LPV	TPV	0.3573	99.8534
NFV	DRV	0.527	99.7646

Table 2.2: Classification statistics for pairs of inhibitors HIVpr using the SWED metric. The Pearson R is between the resistance of the two inhibitors. The 3-fold cross-validated accuracy is shown for random forest.

Inhibitor	Inhibitor	Inhibitor	Accuracy
ATV	DRV	FPV	99.8152
ATV	DRV	IDV	99.8365
ATV	DRV	LPV	99.8519
ATV	DRV	NFV	99.8358
ATV	DRV	SQV	99.8283
ATV	DRV	TPV	99.8183
ATV	FPV	IDV	99.7601
ATV	FPV	LPV	99.7686
ATV	FPV	NFV	99.7516
ATV	FPV	SQV	99.7658
ATV	FPV	TPV	99.7919
ATV	IDV	LPV	99.8149
ATV	IDV	NFV	99.8037
ATV	IDV	SQV	99.7727
ATV	IDV	TPV	99.842
ATV	LPV	NFV	99.7905
ATV	LPV	SQV	99.8075
ATV	LPV	TPV	99.8272
ATV	NFV	SQV	99.7558
ATV	NFV	TPV	99.8016
ATV	SQV	TPV	99.7983

Table 2.3: Classification statistics for a subset of the triples of inhibitors HIVpr using the SWED metric. The 3-fold cross-validated accuracy is shown for random forest.

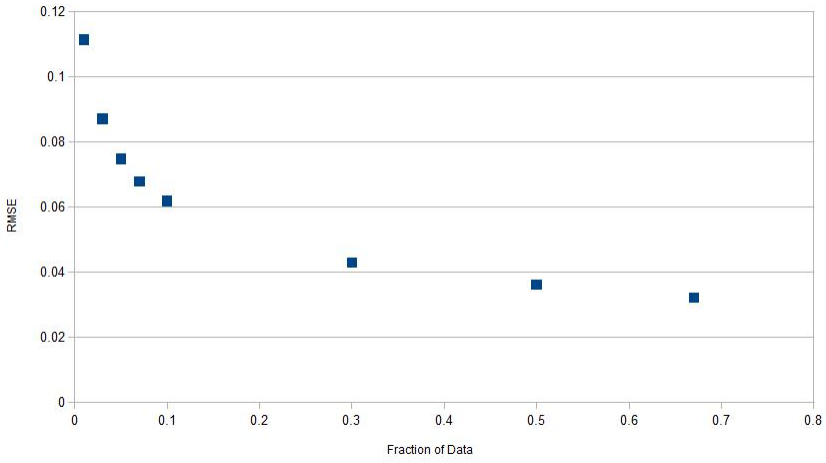

Figure 2.1: The dependence of RMSE on the fraction of data used to train a regression analysis for one inhibitor based on the SWED encoding.

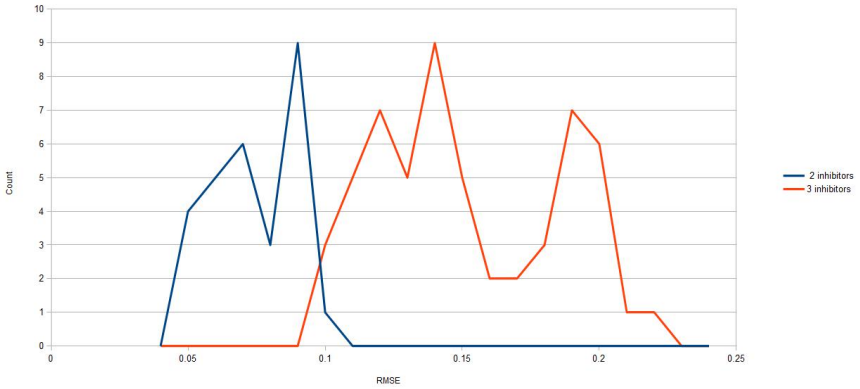

Figure 2.2: The distribution of RMSE between calculated and observed resistance values in 2 and 3 inhibitor regression analyses. For two inhibitors the RMSE ranges from 0.04 to 0.1 and for three inhibitors from 0.1 to 0.22.

HIV. Our software is available from a Github repository [31]. The expanded dataset, even when compressed, was too big for the repository and will be made available upon request to qualified researchers.

Data Expansion The Stanford dataset[27] for HIV protease is comprised of different protease sequences with the observed resistance in the Phenosense assay [28] for the 8 clinical protease inhibitors FPV, ATV, IDV, LPV, NFV, SQV, TPV and DRV. The sequence of the 99-amino acid protease monomer is presented, indicating those amino acids that are different from the consensus sequence of HIV-1 Group M subtype B. Each position in the sequence data may have more than one possible amino acid mutation. These mutations are listed as multiple abbreviations along with insertion * and deletion # for the field of that position. Sequences with two or more alternate amino acids at a single position were expanded by constructing all possible sequences as described in [10],[11]. A total of 1951 genotype sequences were expanded to 414010 single sequences. The expansion potentially could cause "cross talk" where one member of the expansion is in the test set and another in the training set. We have shown previously that this has an insignificant effect[10],[11].

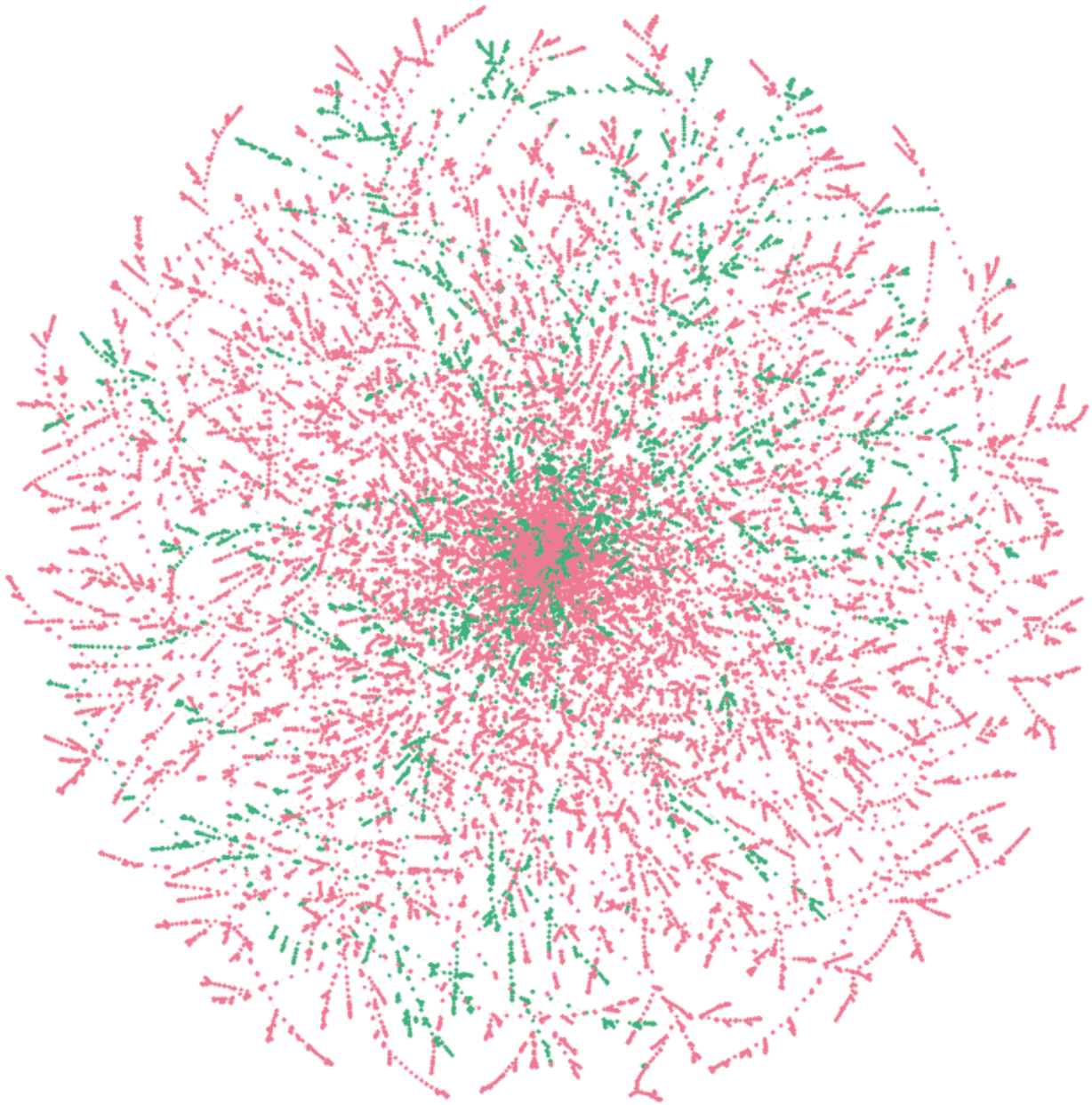

Figure 2.3: The distance based sums(SWED) $l2$ norm spanning trees of ATV resistance. Resistant and non-resistant nodes are represented by green and red colors respectively.

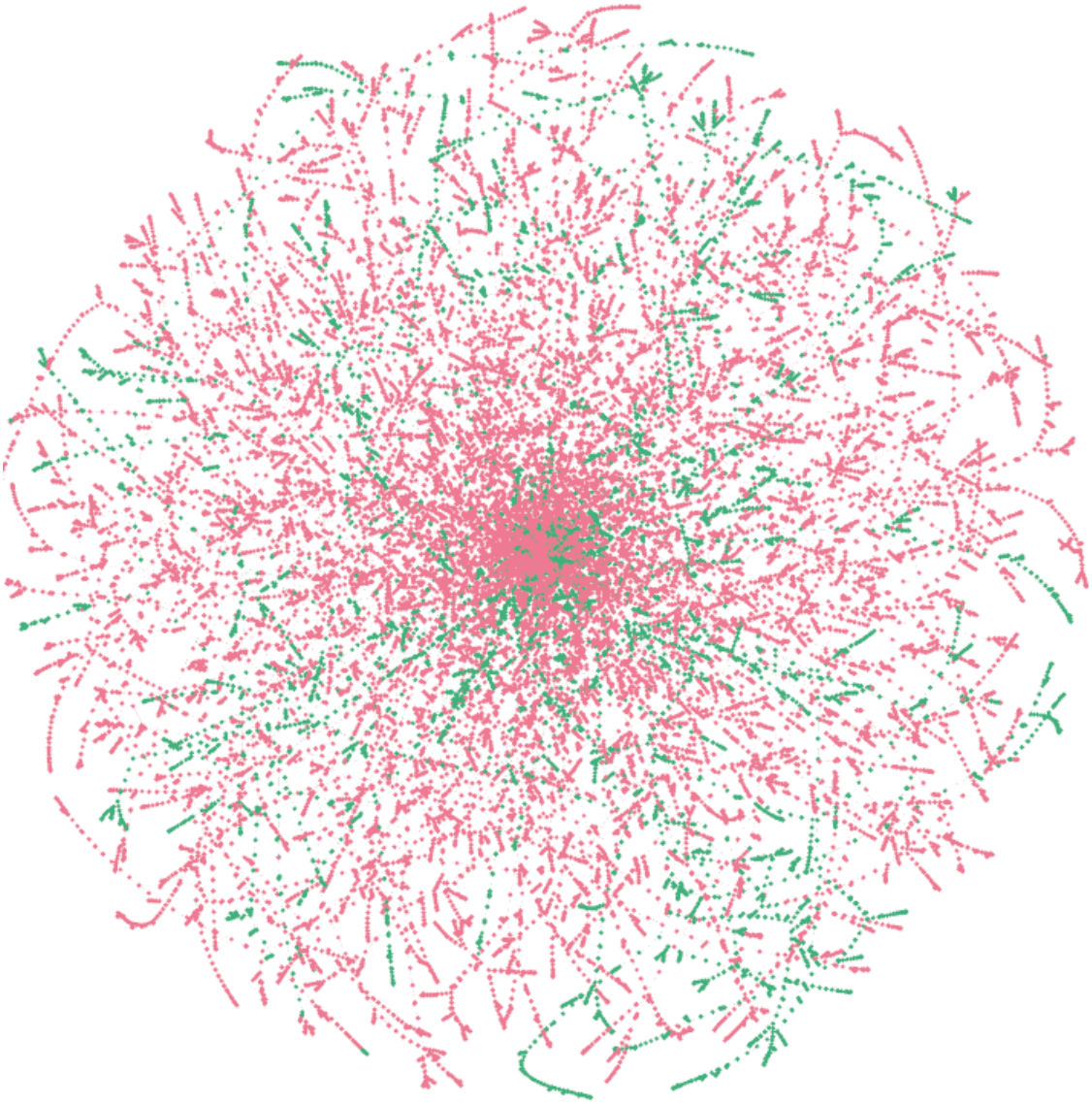

Figure 2.4: The count based sums (RWSED) $l2$ norm spanning trees of ATV resistance. Resistant and non-resistant nodes are represented by green and red colors respectively.

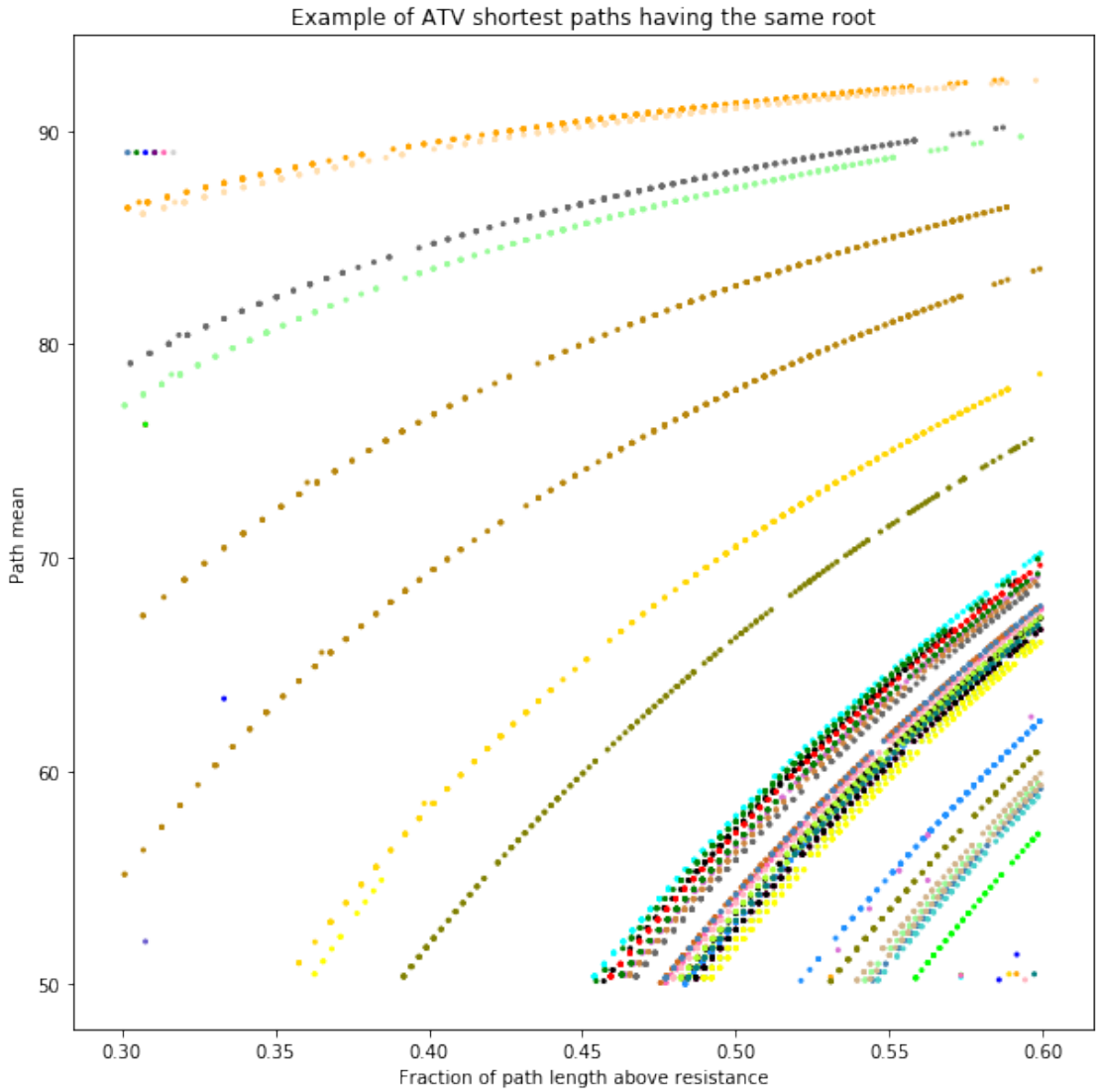

Figure 2.5: Grouping paths with the same root together in a sample of SEWD shortest paths for ATV. The y axis shows the number of paths in each cluster and the x axis shows the fraction of those paths that are above the resistance threshold.

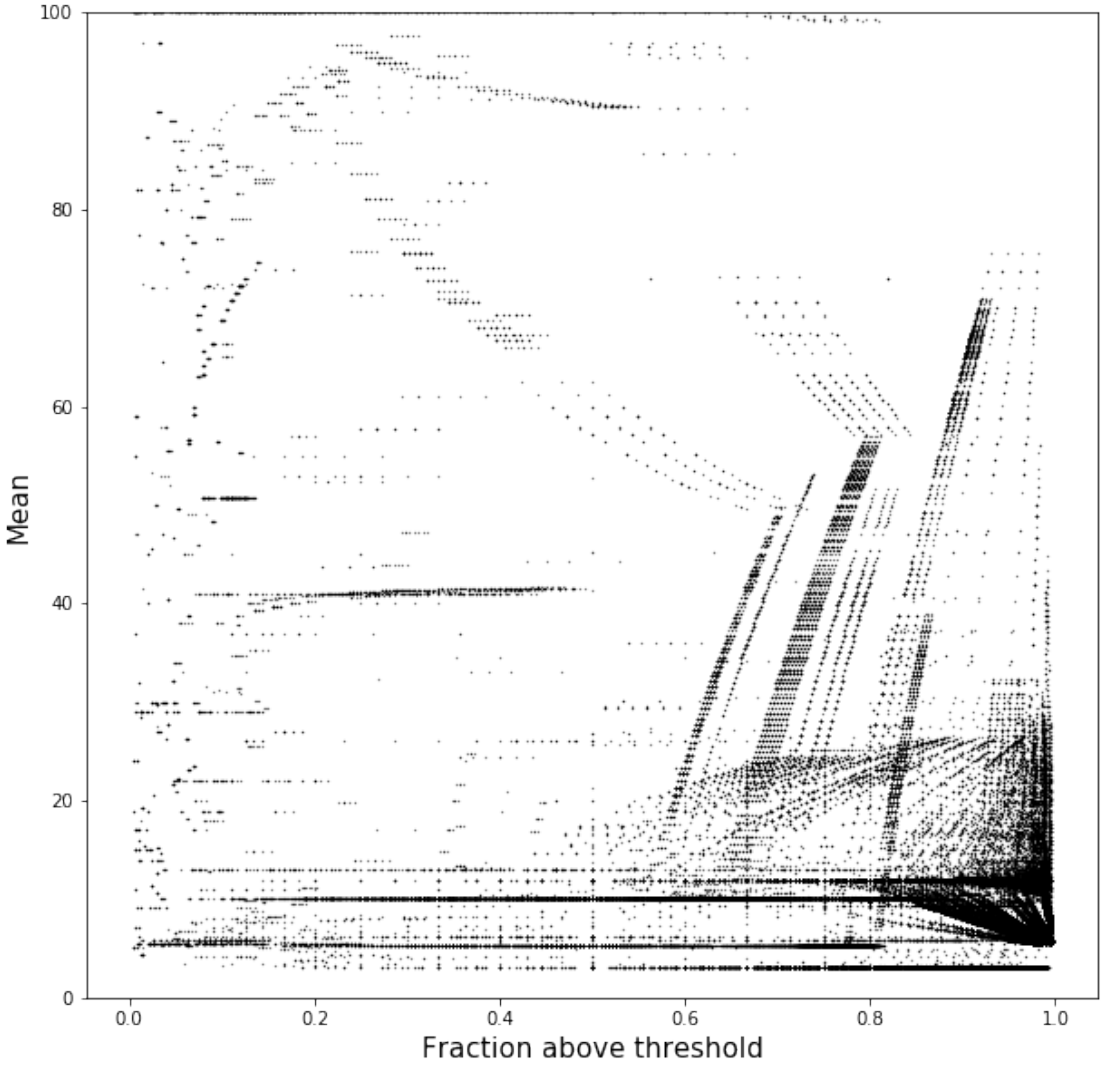

Figure 2.6: SWED shortest paths that gain resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance

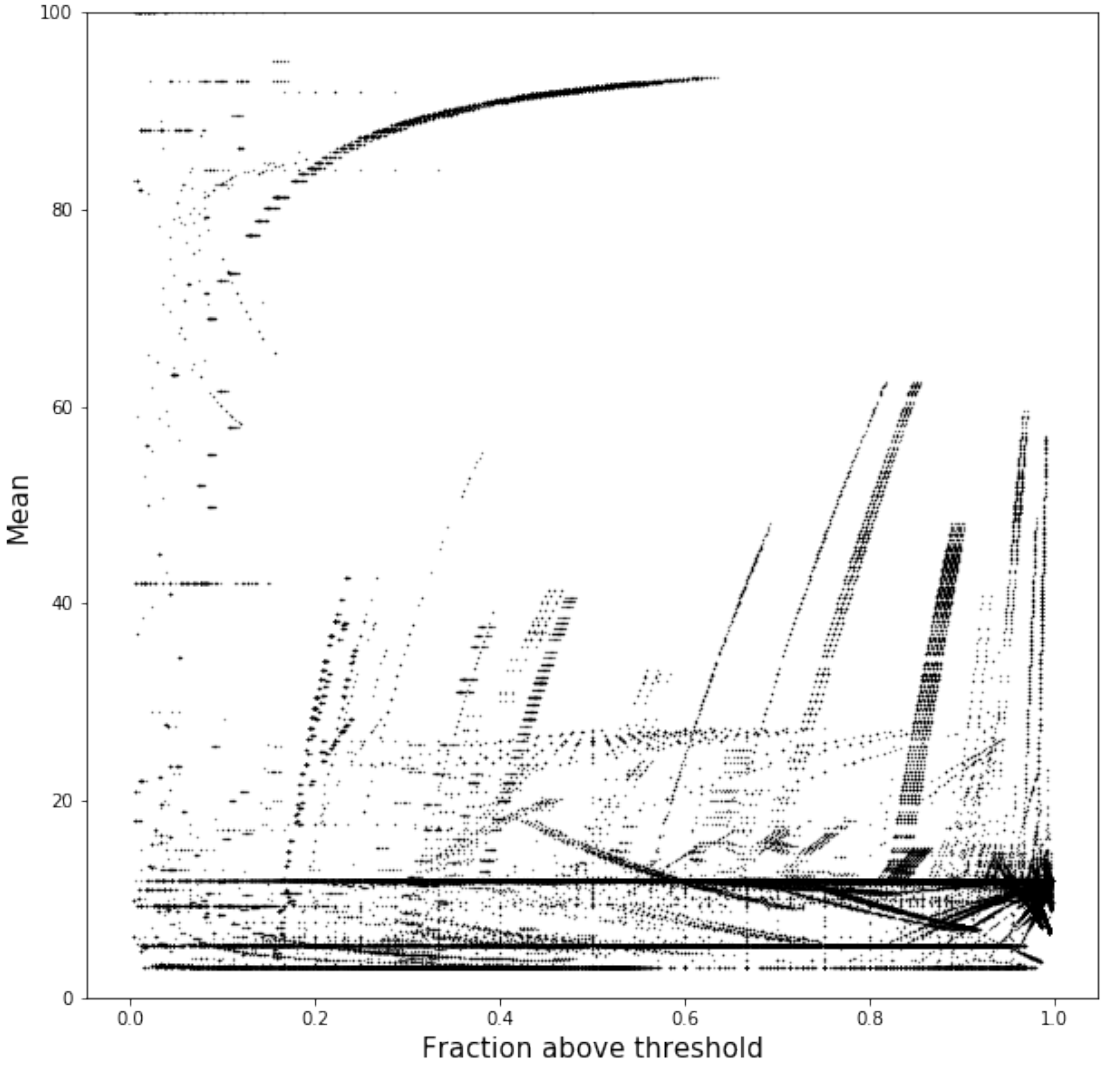

Figure 2.7: RSWED shortest paths that gain resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance.

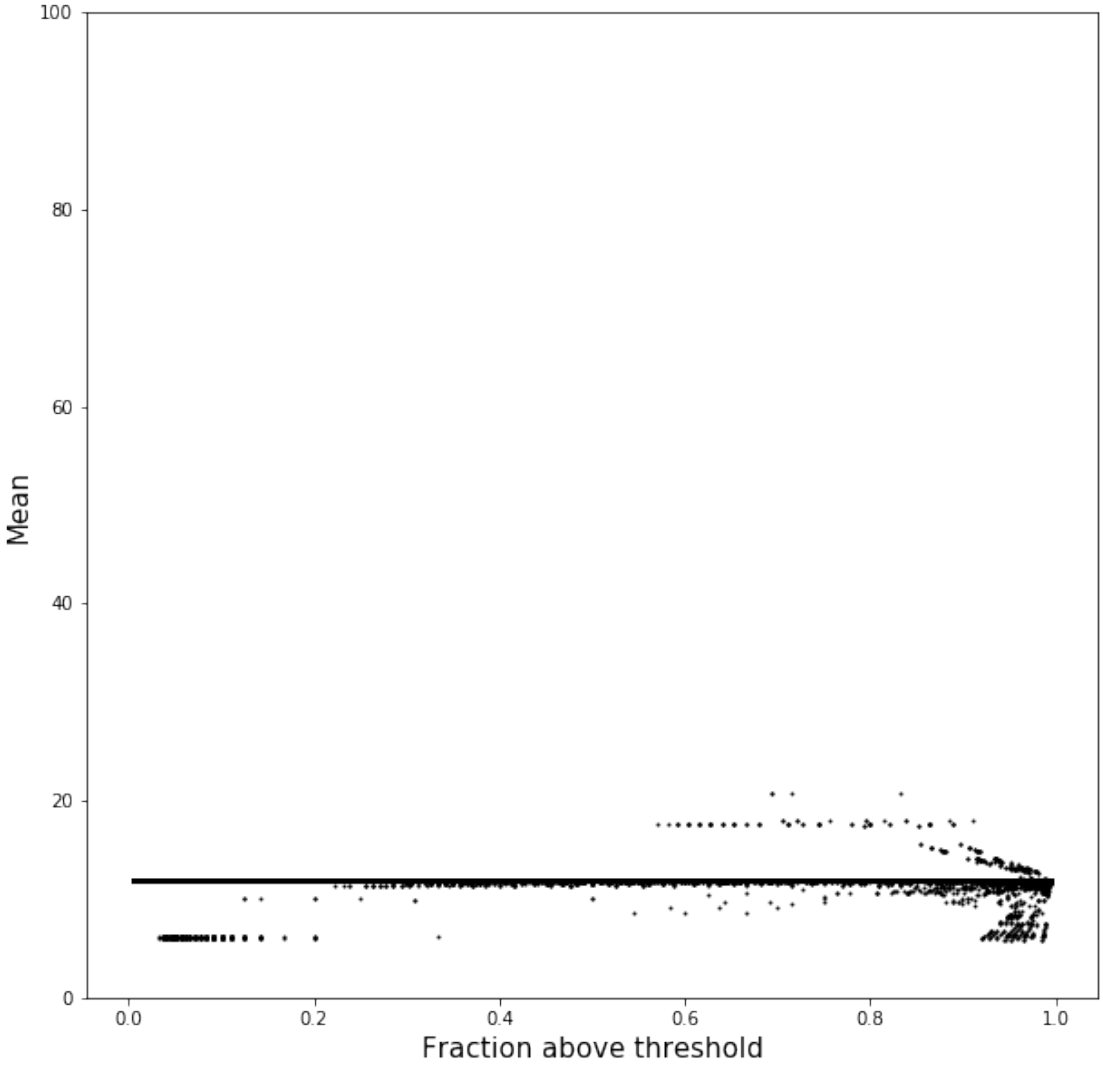

Figure 2.8: SWED shortest paths that lose resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance.

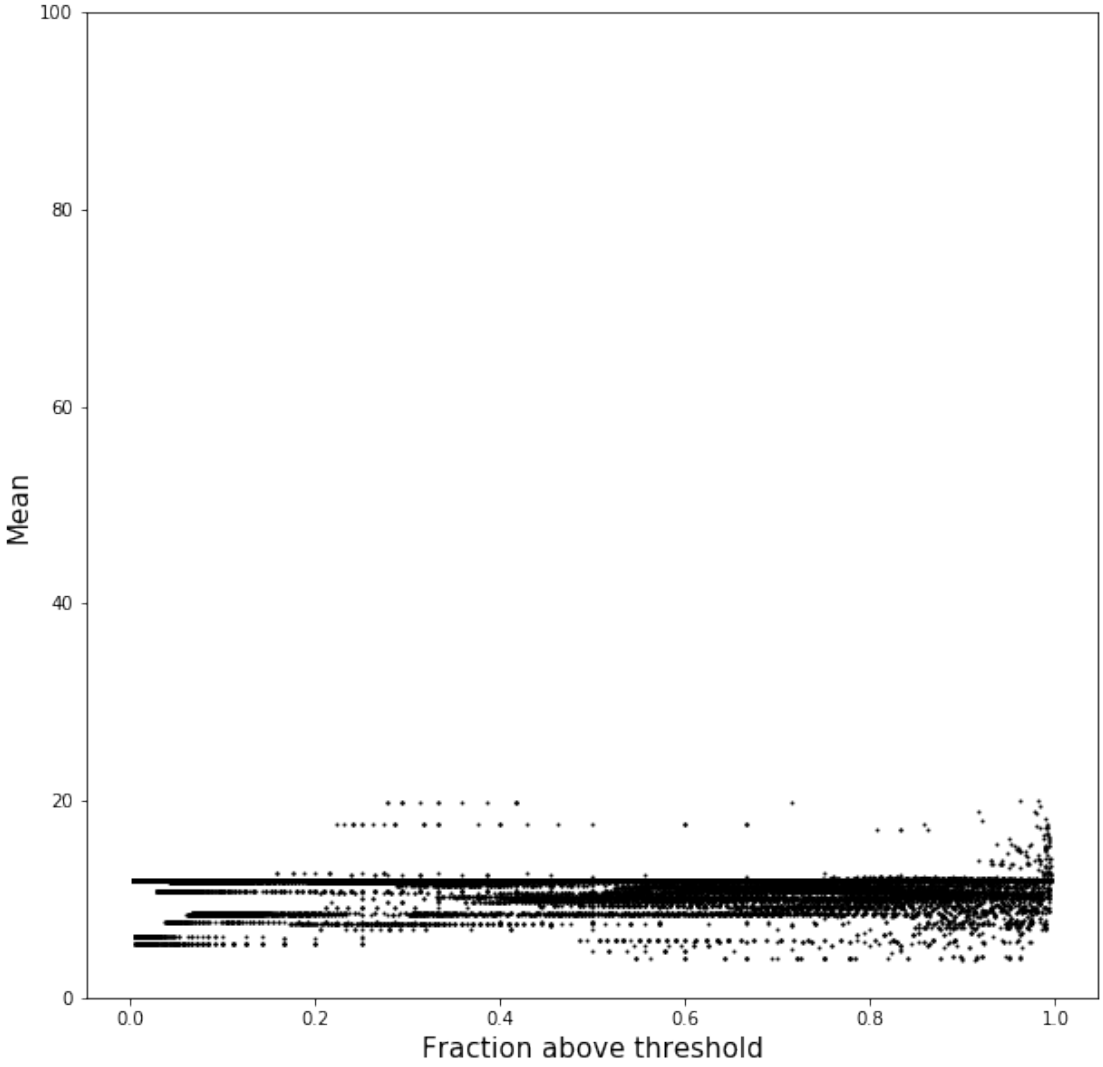

Figure 2.9: RSWED shortest paths that lose resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance.

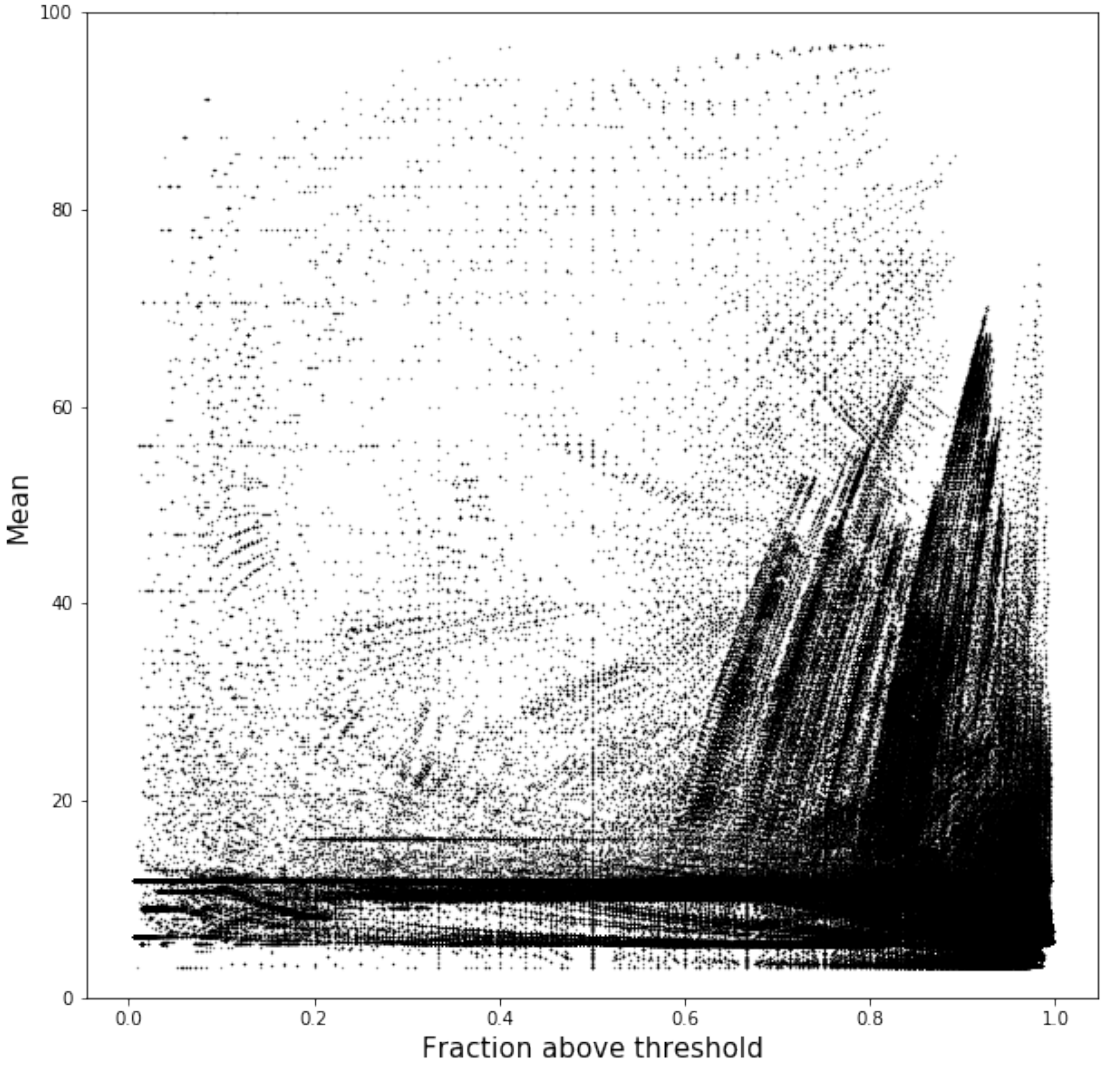

Figure 2.10: SWED shortest paths that fluctuate in resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance.

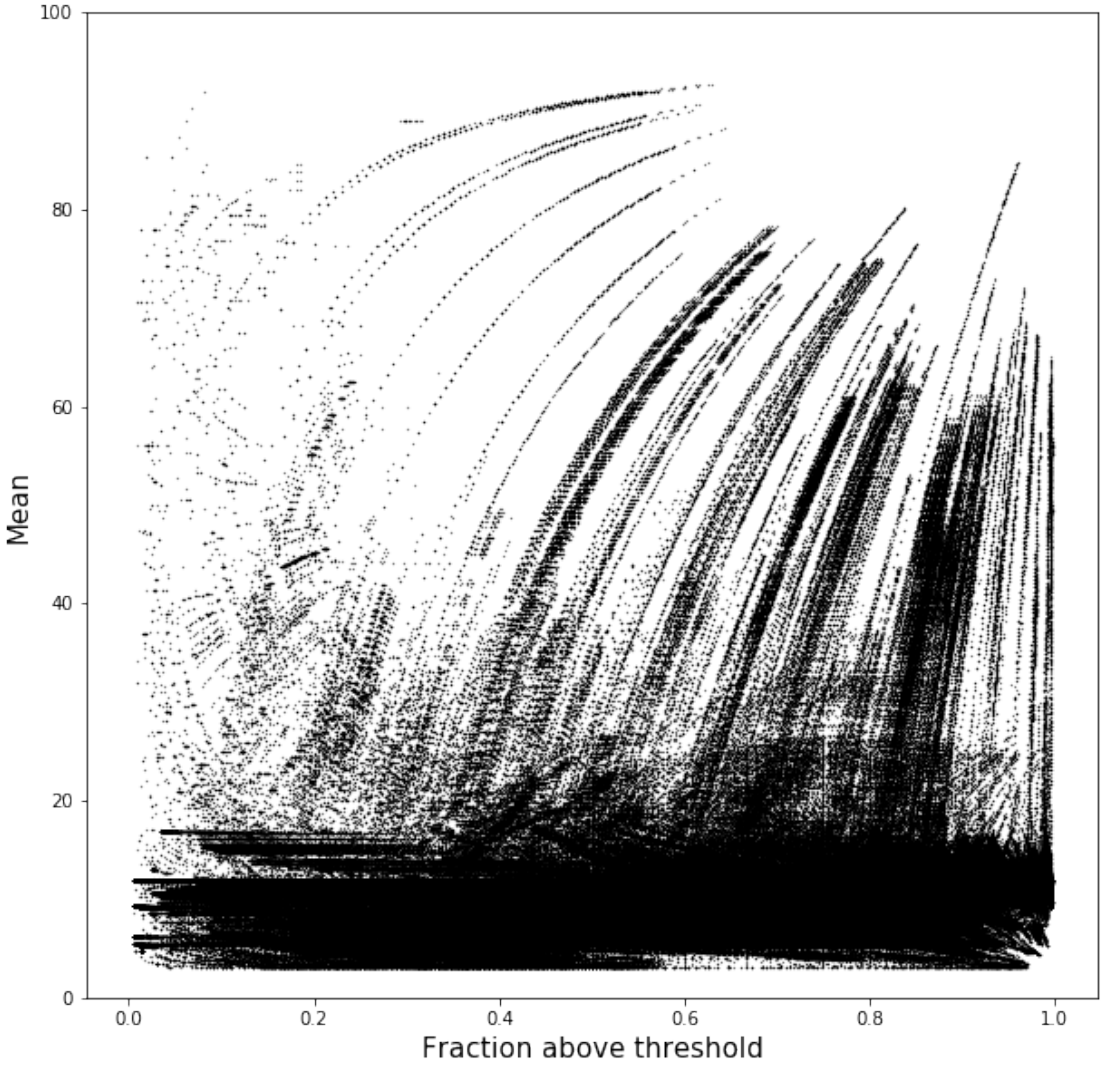

Figure 2.11: RSWED shortest paths that fluctuate in resistance for ATV. The y axis shows the mean value for resistance along the path and the x axis shows the fraction of the path above the threshold for resistance.

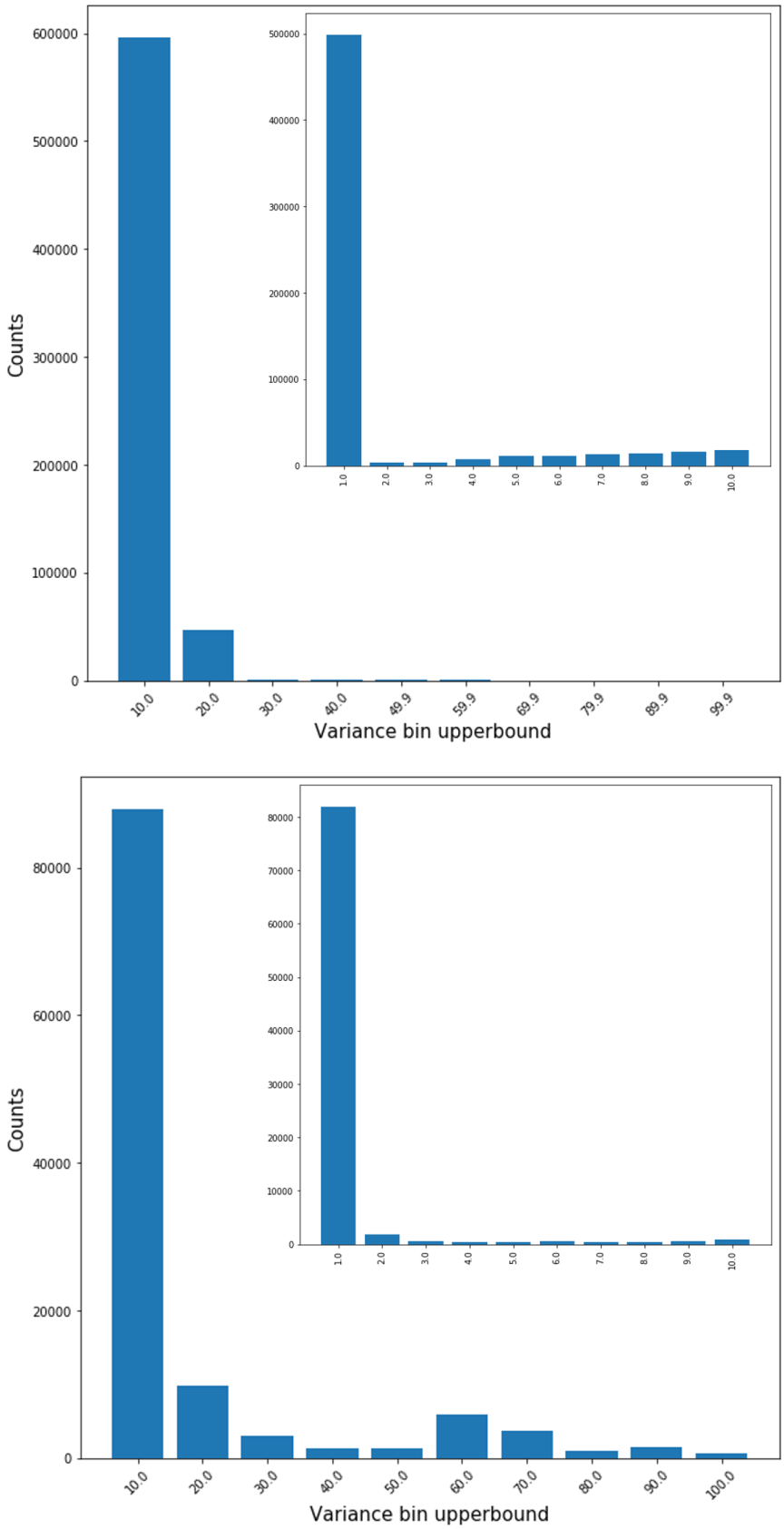

Figure 2.12: SWED(left) and RSWED(right) histograms of the shortest paths that are above resistance for ATV. 1.16% of SWED and 6.8% of RSWED paths have variance greater than 100.The histogram of paths forming the first bin are depicted in the top right corner of each figure.

2.4.1 Vector Generation

Vectors were generated for each sequence by obtaining the neighbors of each position of this sequence from the Delaunay triangulation as was done in [12],[11],[10],[14],[20182018]. The coordinates of the αcarbon atoms were used, and all arcs of the triangulation were used. Earlier studies in our lab [18],[10] showed these were sufficient. The long arcs in the Delaunay triangulation, which correspond to distal surface contacts, are a small subset of the total set of arcs. Other coarse representations of amino acids, such as center of mass, can be highly variable with changes in the kind of amino acid. The first step in this process was to use the positions of each amino acid residue from a crystal structure of the HIV protease dimer with 198 residues (pdb entry 3oxc was used [32]). The Delaunay triangulation was generated exactly once according to the position coordinates obtained from this file and then we obtained the neighbors for each sequence based on this adjacency matrix. A 20x20 amino acid matrix was generated from this adjacency matrix in two different ways: average distance and count between neighboring amino acids. Since this matrix is symmetric, we take the upper triangular values of this matrix as a vector, which is of the size 1x210. The count defines a Structure-Weighted Edit Distance (SWED) and the average distance defines a Radial Structure-Weighted Edit Distance (RSWED).

2.4.2 Classification and Regression

The Stanford database curators recommend a resistance value of 3 in the phenosense assay as the threshold for resistant/non-resistant proteases[27] and we used their recommendation. As a control, since we have recalculated the vectors with new data, the classification calculations were repeated. The values for 3-fold cross validation are shown in Table 1 and demonstrate that the data were generated successfully. The RMSE for regression for one inhibitor as a function of the size of the training set is shown in Figure **??**. This corresponds to a correlation coefficient of $> 99\%$.

In addition to control calculations for single inhibitors, the same calculations were performed for all pairs and triples of inhibitors. The average classification accuracy is $> 99\%$

and the distribution of RMSE is shown in Figure 2.2. Calculations were performed in python using scikit-learn[30]. Regression was done with random-forest regression using two trees. Classification used a linear SVM. Accuracy and F-Score are reported. The F-Score controls for population effects.

$$
\begin{aligned}
Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\
Precision &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN} \\
F - Score &= 2\frac{Precision * Recall}{Precision + Recall}
\end{aligned}
$$

where TP is true positive, TN true negative, FP false positive, and FN false negative

2.4.3 Spanning trees for evolution prediction

Minimum spanning trees were generated for both the SWED and RSWED vectors using Python networkX [33] 2.2 and visualized with Gephi [34] 9.2. However, the amount of data forced us to use a 10 % subset of the data due to limitations of the networkX library. Therefore we repeated the calculation using 10 randomly selected 10 % samples from the data to ensure that the results did not depend on the particular random sample. Nodes with 'NA' resistance values (which were not observed or determined) were removed while making the spanning tree for each inhibitor. Spanning trees were calculated for of each of these splits. Computing spanning trees of the complete graph is computationally expensive and time consuming, hence we used the spanning tree of each split with edges connecting 400 nearest neighbors for each node. Empirically we have observed that this method yields only up to 2% different edges of resulting spanning trees, when calculated 400 nearest neighbors vs complete graphs on these splits.

2.4.4 Shortest paths from roots to leaves in the spanning trees

The roots of this spanning trees are the nodes representing sequences with low numbers of differences from the consensus "wild type" sequence of HIV-1 Group M sub-type B protease. The root nodes are same as or differ by at most two changes from the consensus sequence. We then calculate shortest paths from these nodes to all the leaves in the spanning trees. The spanning trees created by Gephi [34] 9.2 where visualized with Forced Atlas-2 [35] using a layout gravity of 35, node and edge size of 10. We have verified that the visualizations look very similar for all other inhibitors.

2.4.5 Shortest paths classification

As noted in the results, the majority of the shortest paths in these spanning trees have sequences with resistance levels that are not monotone from root to leaves. However, we are interested in the behavior of sequences that gain resistance. Hence we classify the shortest paths in four categories: paths that remain below, paths that remain above resistance level, paths that gain resistance, and paths that lose resistance. We use the direction from root to leaf as the progression for inhibitor resistance values.

2.4.6 Measurement of the resistance variance for resistant path segments

We are interested in the behavior of shortest path segments that are above resistance, namely, how does the resistance level vary when the nodes in the path are resistant. In order to understand this, we calculated the fraction of the path above resistance and the variance of the resistance values for these path nodes.

PART 3

ILLICIT ACTIVITY DETECTION IN LARGE-SCALE DARK AND OPAQUE WEB SOCIAL NETWORKS

Many online chat applications live in a grey area between the legitimate web and the dark net. The Telegram network in particular can aid criminal activities. Telegram hosts "chats" which consist of varied conversations and advertisements. These chats take place among automated "bots" and human users. Classifying legitimate activity from illegitimate activity can aid law enforcement in finding criminals. Social network analysis of Telegram chats presents a difficult problem. Users can change their username or create new accounts. Users involved in criminal activity often do this to obscure their identity. This makes establishing the unique identity behind a given username challenging. Thus we explored classifying users from their language usage in their chat messages.

The volume and velocity of Telegram chat data place it well within the domain of big data. Machine learning and natural language processing (NLP) tools are necessary to classify this chat data. We developed NLP tools for classifying users and the chat group to which their messages belong. We found that legitimate and illegitimate chat groups could be classified with high accuracy. We also were able to classify bots, humans, and advertisements within conversations.

3.1 Introduction

Telegram is a social networking service comparable to but distinct from Twitter. Users of Telegram are semi-anonymous. This anonymity can conceal criminal activities. Examples include identity fraud, drug trade, bank fraud, and animal cruelty. Users also engage in innocuous activity such as Python programming discussions. Unique identification of a user enables attribution of message content to that user. This enables the discovery of

relationships between different activities. User identification and relationship discovery are important efforts in criminology and sociology. This paper applies statistical learning and natural language processing techniques. These are used to create efficient tools for user attribution.

Telegram is a social networking service comparable to but distinct from Twitter. Users of Telegram are semi-anonymous. This anonymity can conceal criminal activities. Examples include identity fraud, drug trade, bank fraud, and animal cruelty. Users also engage in innocuous activity such as Python programming discussions. Unique identification of a user enables attribution of message content to that user. This enables the discovery of relationships between different activities. User identification and relationship discovery are important efforts in criminology and sociology. This paper applies statistical learning and natural language processing techniques. These are used to create efficient tools for user attribution.

This paper aims to develop efficient approaches to identify high-information words in a corpus of text. This allows the separation of one set of text from another with the aim of attributing authorship and general purpose to the texts. The goal of this work is to attribute authorship and group membership rather than to interpret the meaning of the text. Therefore it is important to preserve the idiomatic nature of the texts.

The classic methodology for designing such an NLP classifier is to clean the data for English, create a word dictionary, convert the sentences to one-hot vectors to use a word and a sentence embedding strategy and vectorize text, and use a classifier to classify the texts. Instead, we bypass the embedding by choosing the relatively high information words and creating a feature dictionary with these. We then use binary classification directly on the count hot vectors created by this dictionary. Moreover, the only data cleaning we use here is to replace non-alphanumeric characters by blank spaces and remove stranded/single characters (i.e. words of length 1). We do not remove any words besides these.

We list our contributions as following: implementing a novel word selection strategy that makes for better features, and hence in turn simplifies the classification pipeline for two

text classes. Through this strategy, we can handle much larger volume of test data to be classified by processing the heterogeneous text data in near real time.

3.1.1 Criminology

Previous work by criminologists has shown that dark net users choose encrypted platforms such as Telegram, Signal, and Jabber for communicating with each other [36]. Telegram is a Russian individual and group messaging service that was founded in 2013. The servers hosting the Telegram service are based in the Middle East. As such it is difficult or impossible for U.S. and Western prosecutors to acquire server-side chat records. While the majority of users are engaged in entirely lawful activities a growing criminal element has taken root. Many groups involved in illicit enterprises, which traditionally use the dark web, have found it far more convenient to conduct business over Telegram and similar "grey-web"[37] services. We use the term "opaque" to describe these activities, where users deliberately obfuscate their identities to hide their activities.

Individual groups set up their own channels or chat rooms to deal in their area of expertise. Other groups establish exchanges where services and products may be bartered, for example "sim farms" to fake credentials [38]. User reputation is critical for establishing connections and sales [39]. Therefore establishing computational tools that can track users and their social networks is important [40]. There exists a great deal of overlap between groups, with individual criminals active in multiple groups at any given time as well as the more formal dark web markets.

Within these illicit networks there exists criminal jargon which is unique to the type of crime being committed. Our work focuses on financial and white-collar crime. Thus we frequently see terms like "fullz" (credit card information), or "dumps" (personally identifiable information). These terms are directly related to the types of financial crime being committed in any given group chat.

3.1.2 Telegram Scraping Approaches

Telegram is accessible both through a traditional web browser interface and through their mobile app. An Application Programming Interface (API) is available, however its use is mostly restricted to Telegram-approved bots. The data were scraped by logging into each group and downloading the HTML formatted history of the group chat. The HTML was then parsed as described in section 3.4.3.

3.1.3 Telegram Language

The Telegram data differ from most text data sets in two important ways. First it is highly idiomatic, where the idiomatic language varies from group to group. Second, the messages tend to be short, which raises sampling issues where keywords are missed in a small sample, when the messages are analyzed.

Individual groups employ their own slang, lingo, or cryptolect when they communicate amongst themselves. This is both a measure of inclusion or group membership and a reflection of individual linguistic idiosyncrasies. Groups also can employ their own cryptolect, words, and phrases unique to their criminal activities to intentionally cloak themselves from inspection [41],[42]. "Thieves cant" and Cockney rhyming slang are classic examples of this linguistic process. Traditional NLP approaches, like Word2Vec and Doc2Vec [43],[44], indicate that these terms would be the most feature rich. However many of these methods are based on large, clean data sets of standard English, such as newspaper articles or literary works which limits their applicability with non-standard English.

Telegram messages are typically short with respect to the total vocabulary used in a group. The short message length gives rise to the issue of sample selection. In terms of short sentences most NLP research has focused on sentiment analysis, often using Twitter data [45],[46],[47]. Each message takes a sample of the total vocabulary. When a subset of the words is used in machine learning as a set of features to identify a message source, the messages may not have instances of each member of the feature set. These zeros are relatively non-informative and lead to errors in the machine learning. In our ad detection

we found that messages above a certain length, about 200 words, were all but guaranteed to be ad spam. Since the fraction of words chosen is an arbitrary and adjustable parameter, we chose to use 1%, 5%, 10%, 50% percentiles to select words. 50% corresponds to all the words in the corpus. The accuracy, shown in Table 3.1, improves as a larger subset of the words is used because the expected sampling follows Bernoulli statistics.

3.2 Existing text to vector transformation methods

3.2.1 Word2Vec

Word2Vec is a commonly used method for encoding text for machine learning [43]. In most cases it works better than the earlier bag of words (BOW) approach [48]. However, in the case of small idiomatic vocabularies, the differential in performance is not obvious. Word2Vec can be described as a bag of sacks of words, where each sack contains words that are related to each and describe some concept or idea. When the size of the vocabulary is large, these sacks represent the redundancy in the language. For example, a sack might contain "dog, hound, dogz, dogs, hounds, ..., retriever" which are clearly related. This advantage disappears when the vocabulary is small. Word2Vec will converge to either a small number of sacks, where each sack contains most of the vocabulary, or to a large number of sacks where each sack only contains one or two words. In either case the additional information stored in the features is minimal, but the training of Word2Vec could select the most informative set of words.

Instead we decided to use a version of the BOW approach where information theory is used to identify the interesting words. This is admittedly an approximation to the Word2Vec solution with a small vocabulary, but nonetheless useful. Similar, albeit far more complicated approaches, have been used when implementing bag of features (BOF) approaches to image classification and recall [49]. The difference in complexity is due to the requirement of defining regions in image problems, while in this work the regions are simply the words in the text. The "informed bag of words" defines a set of features which references the content

of the document and corresponds to the words that used the most differently between the two sources. While we did not implement it for this work, the approach is readily extended to phrases and word patterns.

3.2.2 Informed Bag of Words

The derivation of Informed Bag of Words (IBOW) starts from the word frequencies in the data. Let p and q be frequencies or distributions and $classes$ be the number of kinds of objects. The Kullbeck-Leibler divergence $\sum_{classes} plog(\frac{p}{q})$ defines an information distance between p and q. A sum like this is relatively useless for machine learning because it does not identify individual words that can be used as features. Instead we select individual objects, i, where $p_i log(\frac{p_i}{q_i})$ is large relative to other members of the set of objects. These items will be both relatively high probability *and* more likely in p than in q. We refer to these as interesting objects. Similarly $q_i log(\frac{q_i}{p_i})$ will be large when q is interesting. If we select objects that are the extremes of the two-tailed divergence $p_i log(\frac{p_i}{q_i}) - q_i log(\frac{q_i}{p_i})$ then we have a set of objects that maximizes the difference between p and q. As both q_i and p_i must be non-zero in this equation, IBOW works on the intersection of the two data sets.

3.3 Related Work

The dark web and the social networks used for coordination of criminal activities spans many platforms. Research into these networks combines many disciplines. To the best of our knowledge, no research using the techniques we detail in this paper exists. Thus, we provide a broad overview of the current literature that is most relevant to our work.

In [50], Ghosh et al. present a system called Automated Tool for Onion Labeling (ATOL). This system crawls the Tor network to find hidden "onion" web sites to build a corpus of keywords related to criminal activity. The system can then automatically classify the hidden web sites using Term Frequency Inverse Corpus Frequency (TFICF) and a clustering technique similar to k-Means. The clusters are used for "thematic labeling" of the content of web sites. The themes can be thought of as the overall topic of the web site for

which search keywords can be used.

In [51], Tavabi et al. study a large corpus of messages posted to 80 deep and dark web (d2web) forums over a period of more than a year. The study shows how the patterns of discussion evolve and how many forums show similarities in content. Hidden Markov Models (HMM) are used to find latent states between forums. The HMM model also allows for modeling the volatility of the forum content. This is important since the content of the forum may change over time. This could cause the forum to appear to no longer be criminal in nature while still aiding criminal activity.

In [52], Bhalerao et al. propose a graph-based model to discover criminal supply chains. The supply chains are discovered from the English-language "Hack" forums and Russian-language "Antichat" forums. The focus of the paper is commercial postings (similar to the advertisement postings in this paper). An interaction graph of forum activity is built. The graph is defined as $G = (U, E)$, where each node $u \in U$ is a user who posts on the forum, and each edge $(u_a, u_b) \in E$ indicates that user u_a sold a product to user u_b. Forum words are vectorized into their importance ranking using the term-frequency inverse document-frequency (TF-IDF) algorithm. Several classifiers are used for the detection of supply chains.

Finally, in [53], a machine learning model is proposed to classify posts on the Instagram social network related to illegal Internet drug dealing. The authors scrape three months of data from the Instagram web site. A word frequency dictionary is built and used to vectorize the data. Machine learning is applied using a decision tree, random forest, support vector machine, and a LSTM-based deep learning model. Good performance is obtained with all four models.

3.4 Methods

3.4.1 The Telegram Environment

Although there is no official listing of the most popular Telegram groups and bots, there are several online tools that claim to find Telegram groups and channels [54],[55],[56],[57],

[58], [59]. Telegram chat rooms come in two flavors: channels, primarily used for broadcasting admin approved ads or listings, and groups, primarily used for social interactions or chats. We focused on analyzing groups with no or minimal user restrictions on posting, and open to the public, because they provide the stability of tracking them, and have a fair sample of the actors involved in the topic.

These groups organically emerge primarily in two ways: either a virtually well connected person forms them, and invites many of the contacts to join them, or a group of like-minded people who got introduced to each other through common groups form their own group where they have targeted discussions. The participating actors in these groups mostly do not have physical social ties, their purpose of interactions is to either buy, sale or discuss about goods that are considered illegal to trade in most of the countries. Hence, we found that the types of the messages in these groups could be segregated into two main categories: messages that are intended to either look for or sale specific goods/skill sets – "listings", and the rest of the messages – "non listings". We found that the English used in both the types differs significantly. Listings tended to be better phrased and richer in non-alphabetic characters. The non-listings contained English words phrased in non-traditional ways and were shorter in length. However, these Telegram messages differ from popular English used in English speaking countries significantly. This may be because English is not the primary language of most users.

3.4.2 Telegram group selection and network expansion

In order to test our methodology, we picked 100 groups discussing financial fraud ranging from cryptocurrency scams, gift card scams, online service scams, stolen credit card information, and so on. These groups may also contain other types of fraud, but the common intersections of these groups is discussion and information pertaining to financial fraud.

The primary way to search for public groups on Telegram is through its native search feature where a key word search renders up-to 3 groups/users/channels if the keyword is a part of the entity's username string. Since our goal is to understand the groups related to

the broad topic of online money scams, we started with getting 5 groups by searching the keywords such as “fullz” , “dumps” and “CVV”. The criteria of picking and expanding on these seed groups was to make sure that the groups had at-least 50 members, had at least a page of recent English messages, had a public Telegram URL and any member was able to post in the group, and the chat histories of which were accessible form the start of the group.

The only way to advertise about Telegram groups is to post the group links in relevant groups where people can join these groups. Hence, after picking the seeds, we searched for unique recent links of other groups that were posted in these groups and successively kept joining these groups until we hit a link that was not valid. We repeated this process with breadth first search strategy.

Despite of automated tools such as Telegram APIs for a network crawl, we stuck to manual crawling of new groups because we wanted a fair sample of the network, where we refrained joining groups that were posted by a single person in the same group, and did a quick sanity check for users that were recently online and were part of this group, and that the group had indeed relevant messages pertaining to financial frauds. Since our goal was to obtain representative groups across the number of members, balance of posts being all listings to all non-listings, and most of the messages being posted by admins to most of the messages being posted in coherent threads by various people, though we crawled 500+ groups in this process, we only included the ones that would pass these goodness criteria. Upon joining the group, we manually downloaded the chat history of these groups.

3.4.3 Data Preparation

In Telegram messenger’s chat history, each user’s message is saved as an HTML tag with a unique id. However, if a user posts consecutive messages within a few minutes and without any other user posting a message in between, their second and later messages are appended to their first message. We observed that in most cases, when the the text of the first message is merged chronologically with the texts of its appended messages and

considered as a single message block, it conveys a complete sentiment or message in contrast to considering the texts of each of these messages as independent message blocks. Hence, we define a unit message block to be the chronologically ordered texts of a message and its appended messages.

In Telegram chatroom history exports, the users are identified by their non-unique username, and hence we cannot tell the difference between two users with the same username. Hence, for the purpose of this study, we only consider the message block texts as reliable features and ignore all other information from the HTML files. We parsed each group for message block texts. Besides ASCII characters, these messages contain a large variety of UTF-8 and UTF-16 characters, including emojis. However, we focus only on English characters. We cleaned the message text by converting to lower case, replacing non-alphabetic characters by spaces and removing single letter words. For each group, we reserved 1/3rd of messages for testing and hence splitting each groups into 2:1 train:test split by picking message indices randomly, so that the chronological order of the messages becomes insignificant. We then build word counters of each group's training data where we define a word to be space separated set of alphabetic characters. We conclude our data preparation by constructing word Python dictionaries (or hash-tables) which we use to build our models.

By comparing the sizes of train data sets we noticed a huge size imbalance in many cases, and we observed skewed learning in training a binary classifier on training data sets outside of 5 fold range of each-other, hence we only compare the groups whose train data sets are within 5 folds of one another as a preventing measure for skewed learning.

We created/labeled five additional data sets to model questions where NLP techniques are likely to be useful. These questions include distinguishing between bots and normal users, isolating listings, telling legitimate from illicit groups, and distinguishing between different social network agents.

bot data set :

Bots or chatbots are Telegram accounts that are capable of performing specific administrative

tasks, such as printing a welcome message upon joining, printing the user's name change history, getting the Telegram userid, or noting the missing username, periodically posting a message on behalf of admin, removing users if they post specific keywords etc. Many groups in our data set use bots for a variety of tasks. In order to identify if the message was posted by a bot, we extracted 2664 messages posted by 19 users among 13 groups that humans identified as bots.

listings data set :

In order to test if our classifiers distinguish listings from messages, we created a listings data set comprised of adverts for exchanging illegal goods and services. We did so by adding 58179 message blocks – a month's data from a group classified by humans as "listings only" group.

legitimate data set :

In order to test if our classifiers could identify the illicit groups form legitimate groups, we identified a group that discussed constructs of Python programming language and used its 2,2582 most recent message blocks as our legitimate data set.

conversation data set :

In order to test if our classifiers could differentiate between conversations and a mix of conversations and listings, we identified a group with 2664 message blocks whose purpose was to discuss how-tos for network attacks, and used it as conversation data set.

twitter data set :

We wanted to measure how the language of Telegram differs from that of the contemporary popular English. In particular, can our models distinguish between current twitter English and Telegram English? In order to test this, we picked English tweets for a random day in past 1 year and randomized the tweets to create this data set.

3.4.4 Language Model

In order to evaluate the language model, we defined a series of test systems. The first and primary test was to see if the model could discriminate between one Telegram group and another. Since we had identified and scraped 102 different groups, this gives 5253 unique combinations of groups which is enough to derive meaningful statistics for the quality of the model. We evaluated pairs of groups to maintain balanced or nearly balanced data sets. It typically took about three hours of time to run five fold cross validation on all unique pairs without GPU enhancement with a 1% percentile cutoff for the words and twelve hours with 50% (which used all the words).

The second test was to distinguish between regular human users and bots or automatic users. We manually extracted bot messages and compared them to random selections of human messages.

The third test was to compare advertisements or listings with conversational messages. We assayed this both by comparing groups that were almost entirely listings with normal groups and extracting an "ad" data set and comparing that with comparably sized random selections of conversations.

Fourth, we compared a Python programming group, which is a sample of non-criminal activity, with our "illicit" groups to show that we could discriminate between licit and illicit activity, and finally, we compared Telegram messages to Twitter messages.

After creating word counters from the training data for each of the data sets, we compared all groups among the same data-length component of all the groups falling into 5 fold range of the size of their training data. for the cases of legitimate, bots, listings and twitter data set, we compared as many lines of the smallest data set. Given two groups, the goal here is to identify relatively high entropy words that make better features. In order to achieve this goal, we paired all groups with other unique groups and compared all the common words of the pair by calculating relative entropies. We then picked 1, 5, 10, 50% of words on each end of the distribution generating three lists of high entropy words. We used these words as the dictionary words for encoding each message block, thereby creating four

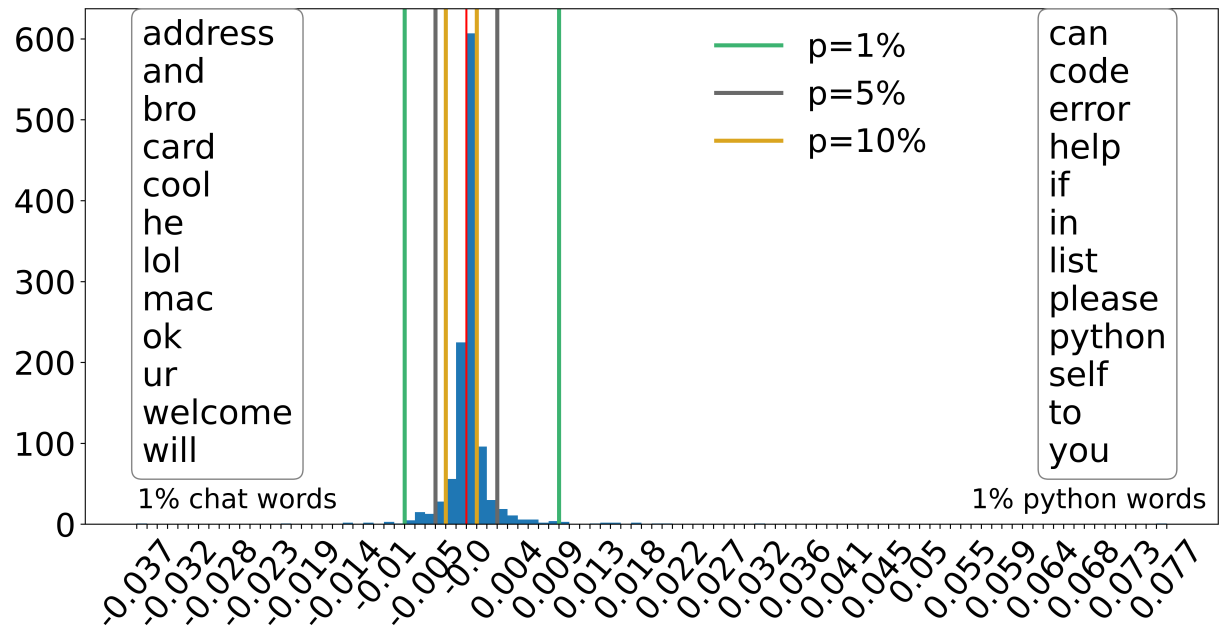

Figure 3.1: A histogram of the relative entropy vs. count for the Python group and an illicit group. The words shown are those that are significant at the 1% percentile cutoff. The information metric has clearly identified the difference between the two groups. The red vertical line shows the median.

different types of train-test vectors. For each pair of groups, we then trained the same classifier with the same hyper-parameters for each of these train vectors to compare how much fluctuation in accuracy, precision, recall, sensitivity or specificity, F1 score, and Matthew's correlation coefficient was observed.

3.4.5 Encoding Features

Embedding techniques like Word2Vec [43], Doc2Vec [44], and Paragraph2Vec [60] project the words in a document onto a set of vectors of similar words. Unfortunately, the small size and idiomatic nature of the vocabulary in the Telegram messages tends to obviate the advantage of this approach. The word vectors would be much shorter than the 100 or so typically used in Word2Vec to represent a single concept. Bots, for example, often have a total vocabulary that is much smaller than 100 words, so reducing them to typical Word2Vec sizes is impractical at best. Similarly, consistent misspellings and non-standard

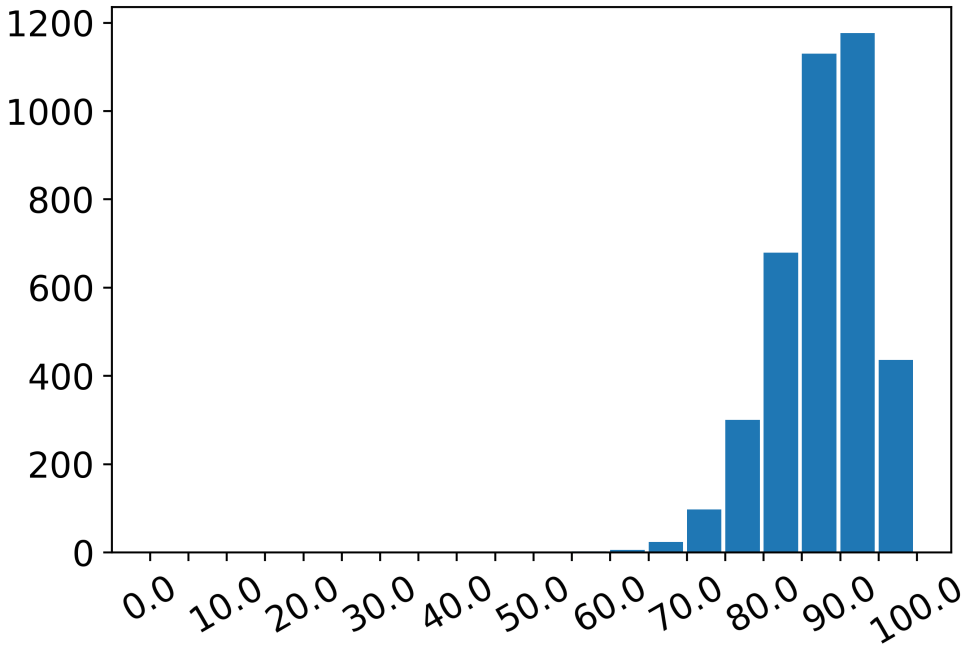

Figure 3.2: A histogram of the accuracy vs. count for classifiers of all distinct group pairs built with 10% percentile of the words. As expected, it neatly follows a β distribution.

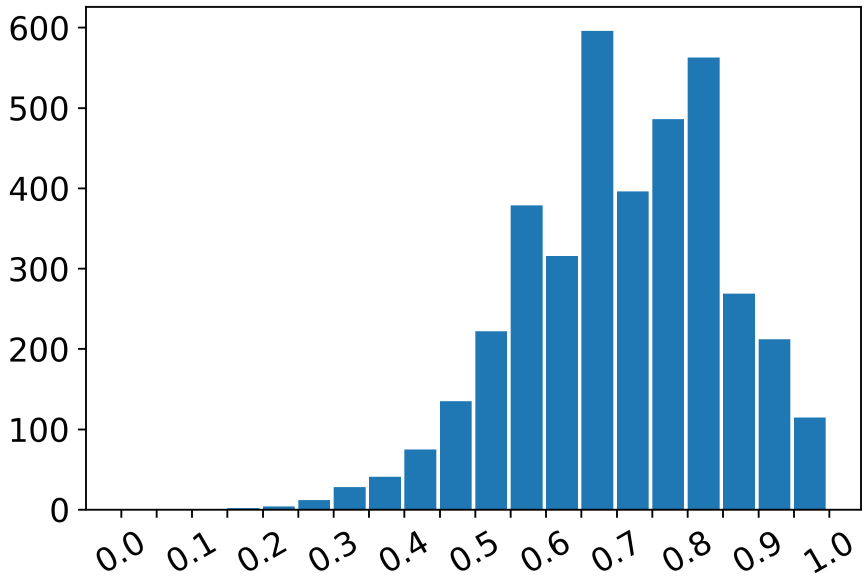

Figure 3.3: A histogram of the Matthew's correlation vs. count for classifiers of all distinct group pairs built with 10% of the words.

usage is characteristic of some of the Telegram users. Consistent use of "ur" instead of "your" is a useful feature for identifying a user or group, which the vector of words, alternate spellings, and synonyms in Word2Vec would hide.

In order to vectorize the message blocks, we first take the p=1, 5, or 10% words from the word-entropy distribution. We use these words as dictionary, and construct ordered word-count for this dictionary for each of the message blocks. Hence, each message block is transformed to a $1 \times |D|$ vector where $|D|$ is the length of the dictionary. And if i^{th} positions of this vector is nonzero, say j, then this message block has i^{th} word in the word dictionary appear in it j many times. The sparse number of words in the word dictionary assures that most of the message blocks would correspond to zero vectors, and hence the training matrix would be sparse. Also, the smaller the p, the shorter the word-dictionary and the sparser the training matrix.

3.4.6 Control Calculations

In order to measure the effectiveness of IBOW as a sentence embedding, we compare IBOW with Doc2vec [60] sentence embedding and TF-IDF [61],[62],[63]. For Doc2Vec we treated each message block as a document and provided a document id as the message block number. We ran Doc2Vec with the gensim [64] library implementation with distributed memory for 200 iterations, with linear decrease of learning rate from 0.03 to 0.01 and vector size 52. We used sci-kit learn's [65] TfidfVectorizer with l2 normalization and without a threshold to build the TF-IDF model in this study. Thus in our study the TF-IDF model used far more features than the IBOW model (see Figure 3.4).

3.4.7 Experimental Details

With word dictionaries for p = 1, 5, 10 and 50, we vectorized training message blocks as ordered word counters for each pair of the groups. To study the goodness of this feature selection method, we train three different binary classifiers for the same training vectors: logistic regression and 1 layered artificial neural network (ann) capped by 200 iterations and

adaboost with 100 solvers. For ann, we use 100 neurons with stochastic gradient decent and constant learning rate 0.01. None of the hyper-parameters are tuned for any specific group-pair, pr threshold values, rather, the hyper-parameters are chosen to be smallest but generally optimal for many groups. We conducted the experiments with Python 3.7 on an 8core double-threaded i7 3.6GHz machine. We used numpy 1.18 [66],[67] with random seed 0 to process the data and scikit-learn 0.23 [65] with random seed 0 to build the models. The Matthew's correlation coefficient is a good alternative to F1 scores and ROC curves [68] for representing training quality. The accuracy and Matthew's Correlation coefficient are defined:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$MC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

3.5 Results

Our results are comparable with state of the art algorithms on these data. Table 3.1 shows the mean and standard deviations as a function of algorithm and percentile. Figure 3.1 shows an example of the distribution of words for Python vs an illicit group with respect to the relative entropy measure. Most of the words near the median are not informative, and even the 1% percentile shows meaningful differences in the features. It is clear from the words shown in Figure 3.1 that the Python group is mostly concerned with helping solve programming errors and the other group about credit cards. Figure 3.2 shows the distribution of the accuracy and Figure 3.3 the Matthew's correlation coefficient when all groups are trained against all other groups in the data with a 10% percentile with an ANN. The TP rate vs FP rate is shown for a typical criminal group in figure 3.7. This figure shows the effect of percentile of data used on the reliability of the machine learning model. In order

to summarize the results, the histograms show the counts over all 5253 unique pairs. The distributions follow a β distribution, which expected for random errors when the minimum and maximum values are constrained to a range. Generally speaking there is not a large difference between using the 10% percentile and the 50% percentile. Using more data in the model, not unsurprisingly, improves the results, but most of the improvement is seen early on.

The Figures 3.6, 3.9, and 3.11 show our the accuracy of our results on different kinds of data. The corresponding Figures 3.8, 3.10, and 3.12 show the corresponding Matthew's correlation coefficient. Five-fold cross validation was run on many individual runs and the figures show a histogram of the relevant statistic. Since the choice of percentile cutoff is arbitrary each row in the figures shows a different value of the percentile. As expected, the distributions have less spread and move towards the right as the fraction of data used is increased.

3.5.1 Comparison with TF-IDF and Doc2Vec

TF-IDF is based on a probabilistic model of word occurrence in text [63],[62]. There have been efforts to place it on an information theoretical basis [61] and the information theoretical basis resembles the IBOW approach. The major difference between IBOW and TF-IDF is that TF-IDF uses the Kullback-Leibler divergence for the probability of the document given the word, while IBOW uses a two-tailed Kullback-Leibler divergence for the relative probability of the word in two sets of documents. In its classic formulation TF-IDF identifies weights based on word frequencies corrected for the frequency of the word in a set of documents. These weights are then used as a linear discriminator for classification. As such TF-IDF uses the union of the words in the total corpus as a feature set unless a threshold is set. The threshold corresponds to IBOW's choice of percentile.

Comparison of the accuracy of IBOW with TF-IDF and Doc2Vec is shown in Figure 3.5. Both TF-IDF and IBOW are significantly better than Doc2Vec, and TF-IDF, as run, is slightly better than our approach. That TF-IDF is better is not surprising as TF-IDF

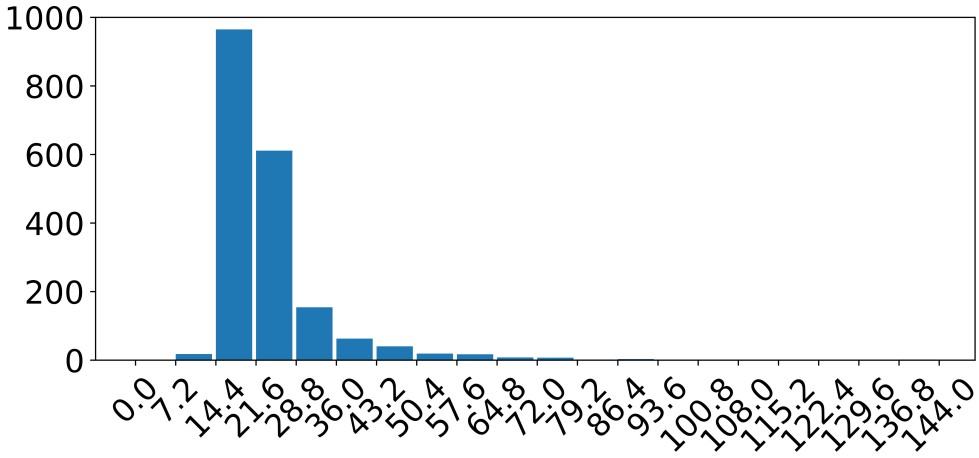

Figure 3.4: A histogram of the number of TF-IDF words divided by the number of words in our approach. The data for the 10% percentile in IBOW is shown.

uses far more features than IBOW as shown in Figure 3.4. IBOW generates nearly the same accuracy as TF-IDF with at least ten times less features which shows that its linguistic model effectively captures the essential feature of the conversation. Using so many more features to receive a few percent better accuracy violates the economy of features which is important for performance with big data. Doc2Vec, in addition to being less accurate than either TF-IDF or IBOW, is significantly slower and averages 20 or more CPU minutes per trial where the other approaches use less than a minute. These data are summarized over a large (1689 samples) and representative subset of the one group vs another test set. The CPU requirements for Doc2Vec limited the size of the test. The machine learning was done with Logistic Regression to eliminate the effects due to differences tuning meta-parameters in Adaboost and ANN.

3.5.2 Bot Detection

Bot messages vary between highly repetitive messages, such as “Welcome to the group” and more complex messages where the group administrator uses the bot to post an announcement. However, even in the more complex messages, the language is highly formatted and relatively rigid with respect to human-based chats. The distributions shown in Figures 3.6 and 3.8 demonstrate that reasonable accuracies can be found even with fairly small numbers

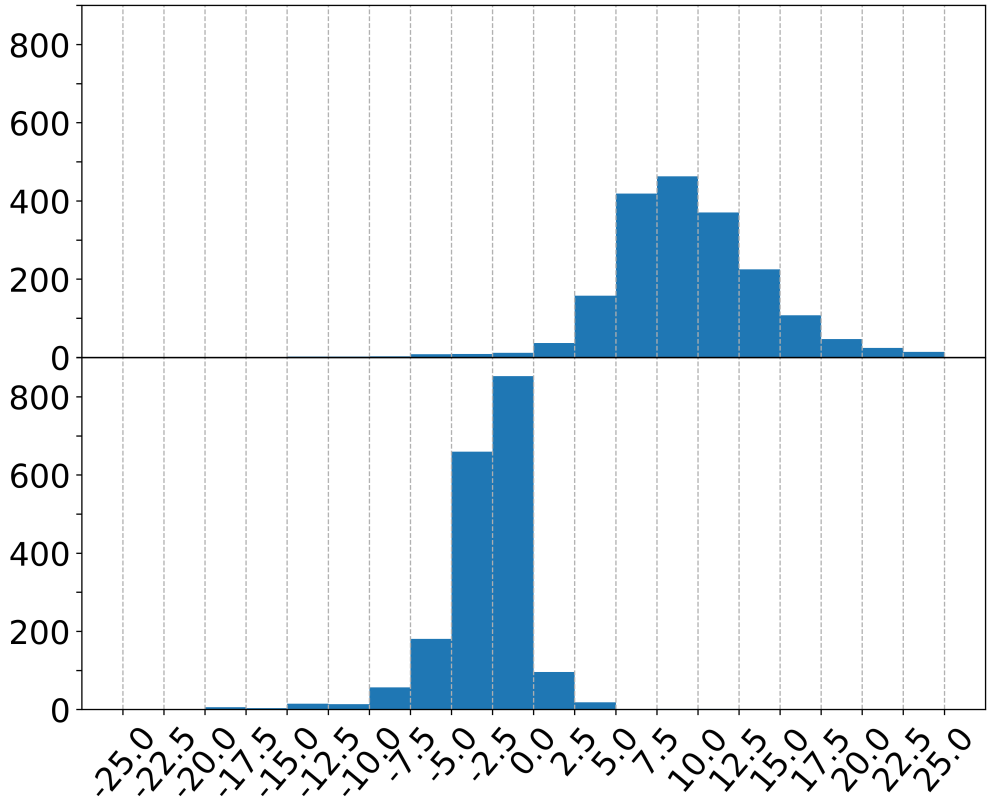

Figure 3.5: A histogram of relative accuracy of our method (10% percentile). The upper panel shows IBOW-Doc2Vec and the lower panel shows IBOW-TF-IDF. Positive numbers are where IBOW is better.

of words (1% percentile). The Matthew's coefficient shows that using more words improves the balance between FN and FP predictions.

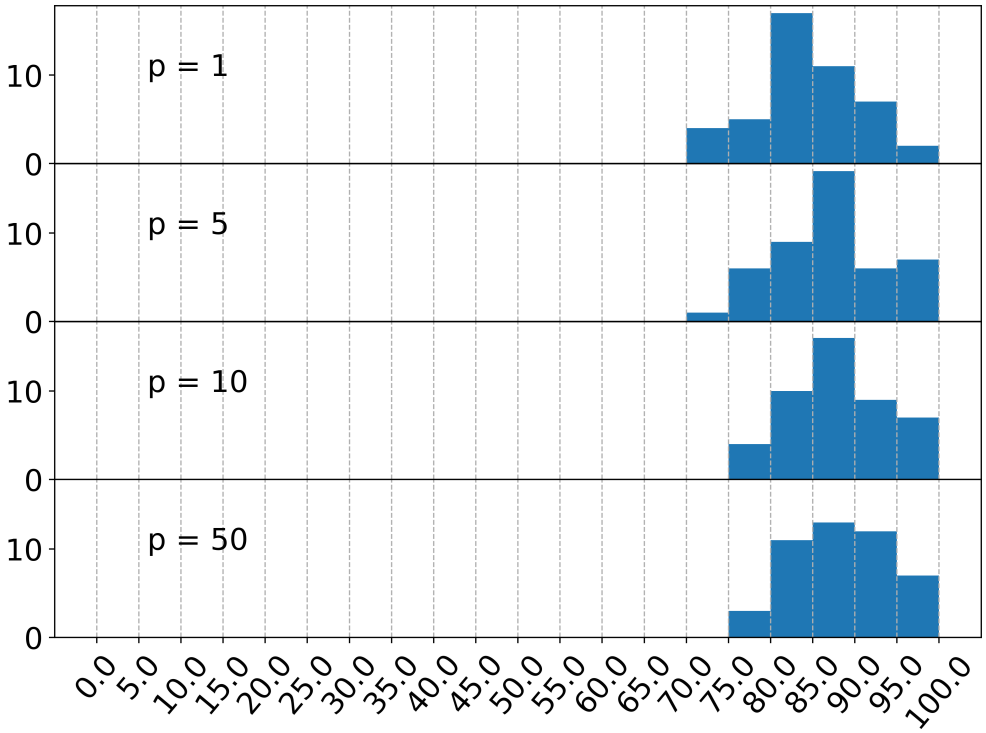

Figure 3.6: A histogram of the cross-validated accuracy of the deciding whether a message came from a bot or a human. Each row shows the dependence on the percentile of the data used.

3.5.3 Listing Detection

Listings or ads are less repetitive than bot messages. Since they are written by members of the community they more difficult to distinguish from chats because the linguistic features of a listing and a chat written by the same person will be highly similar. Figure 3.9 shows the distribution of accuracy and it is not as good as the bot example (Figure 3.6). The Matthew's correlation, Figure 3.10, shows a wider spread than that seen with bots as well.

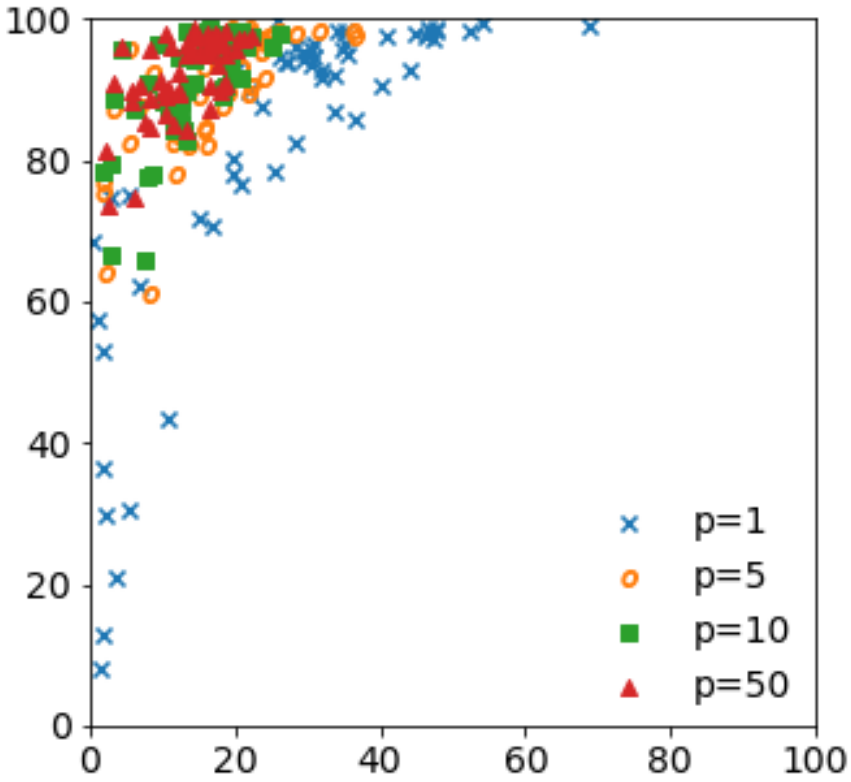

Figure 3.7: A scatter plot of a typical criminal group with respect to other criminal groups. The data of 1, 5, 10 and 50 percentile with false positive messages as x axis and true positive messages as y axis, obtained by artificial neural network. The figure shows that the reliability of the model increases as the percentile of data is increased.

3.5.4 Group Comparisons

The overall comparison of groups, all 5253 pairs, is shown in Figures 3.2 and 3.3. More interesting, from a sociological perspective is the accuracy at determining whether a group was one centered on bank fraud or other illicit activity or one centered on a normal activity. Since many of the Telegram groups have misleading names, for example one on "puppies" being images of dog-fighting, we chose a Python programming group as a control set. We could quickly verify that the Python group was on programming and not snakes. Another similar control was to establish if a message came from Telegram or the similar social networking

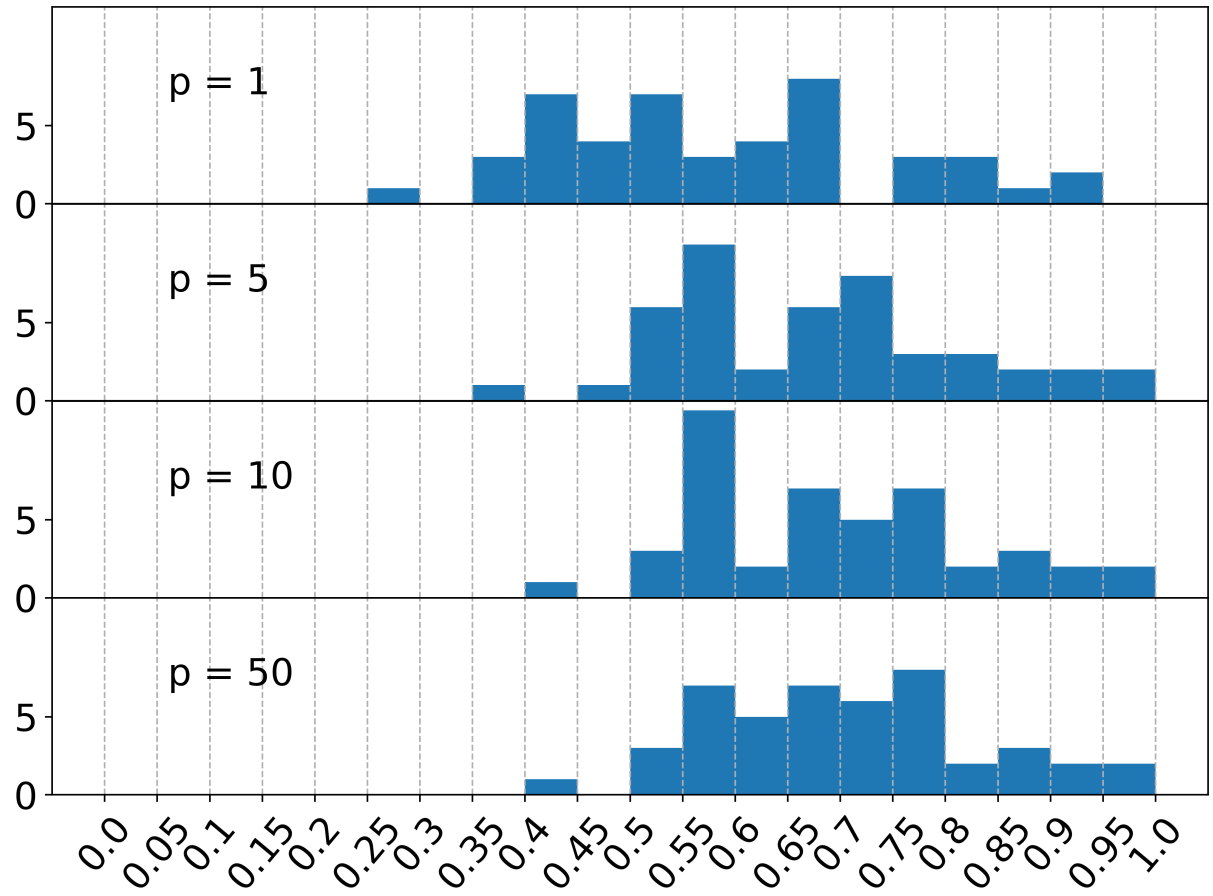

Figure 3.8: A histogram of the Matthew's correlation coefficient of the deciding whether a message came from a bot or a human. Each row shows the dependence on the percentile of the data used.

site, Twitter.

Python vs. Illicit :

Figure 3.1 shows the distribution of words for Python vs a similar sized bank fraud chat group. Interestingly, even at 1% percentile, the words show a human-comprehensible difference. This demonstrates that IBOW can be used to develop human comprehensible machine learning models. The Python group, in addition to the word "Python", includes Python language terms (e.g. "self", "list") and words that describe the kinds of conversations ("help", "code", "error"). The words in the Python group are clearly literate and avoid slang. The bank fraud group has distinctive slang ("bro", "lol", "ur"), and content ("card", "address",

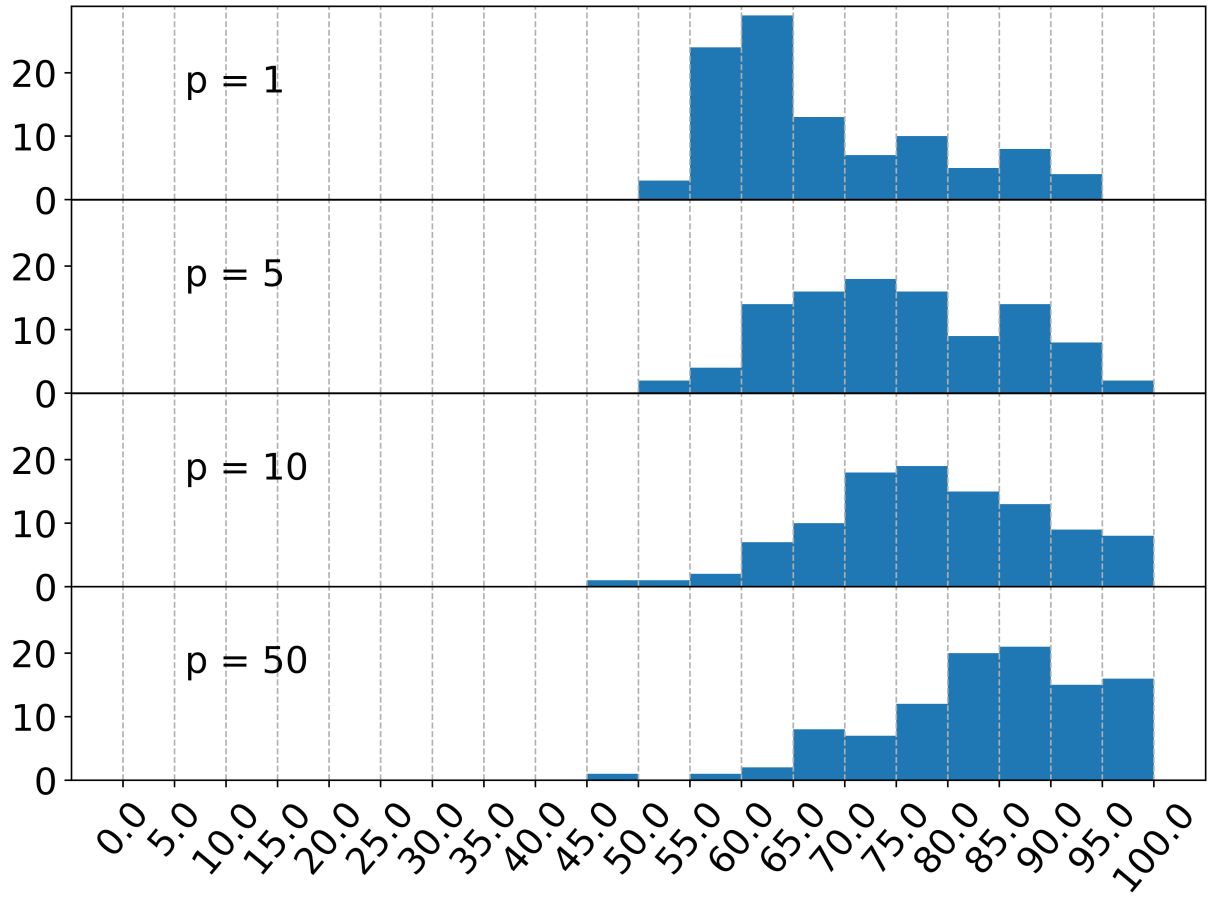

Figure 3.9: A histogram of the cross-validated accuracy of the deciding whether a message was a listing or a conversation. Each row shows the dependence on the percentile of the data used.

"mac"). The two groups also differ in their use of "low information" words like "and", "you", "in", and "ok". Figure 3.11 shows the distribution of the accuracy and Figure 3.12 distribution of the Matthew's coefficient for Python vs illicit chats. Again, increasing the number of words has a larger effect on the Matthew's coefficient than the accuracy.

Twitter vs Telegram :

Figure 3.13 shows the distribution of accuracy when Twitter data were compared to Telegram data. The quality of classification was much higher than with Telegram vs Telegram, so we only present the 10% percentile here. All tweets from a random day, September 23 2019, were downloaded from twitter dumps. Tweets were selected that were in English, more than

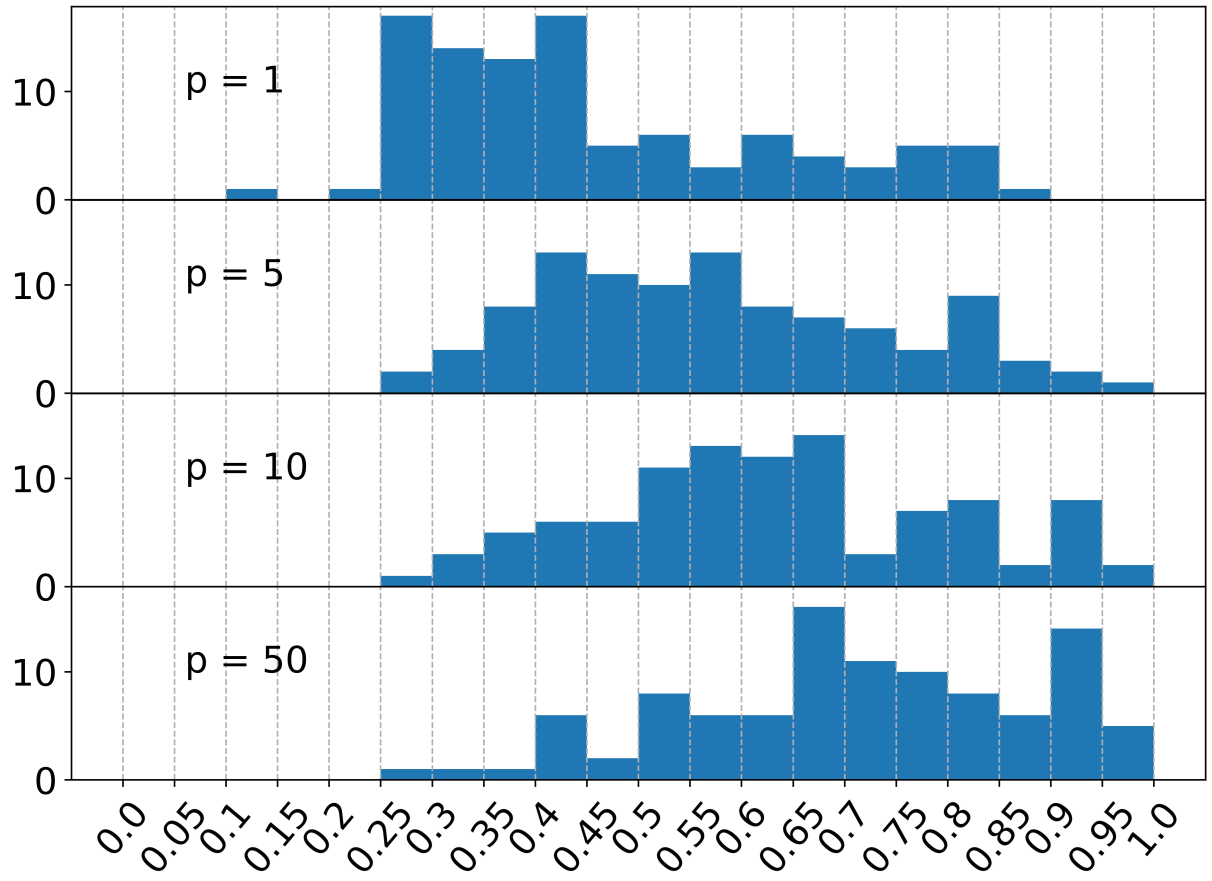

Figure 3.10: A histogram of the Matthew's correlation coefficient of the deciding whether a message was a listing or a conversation. Each row shows the dependence on the percentile of the data used.

two million, and processed in the same manner as the Telegram data. Random balanced subsets of the processed Twitter data were used to ensure that the machine learning reflected the accuracy of the model rather were due to imbalanced data.

3.6 Conclusions

3.6.1 Summary

Our work shows that a relatively simple and economical algorithm, based on a two-tailed Kullback-Leibler divergence, determines a set of features that accurately classify short idiomatic messages. The algorithm is economical both in memory use and computer processing

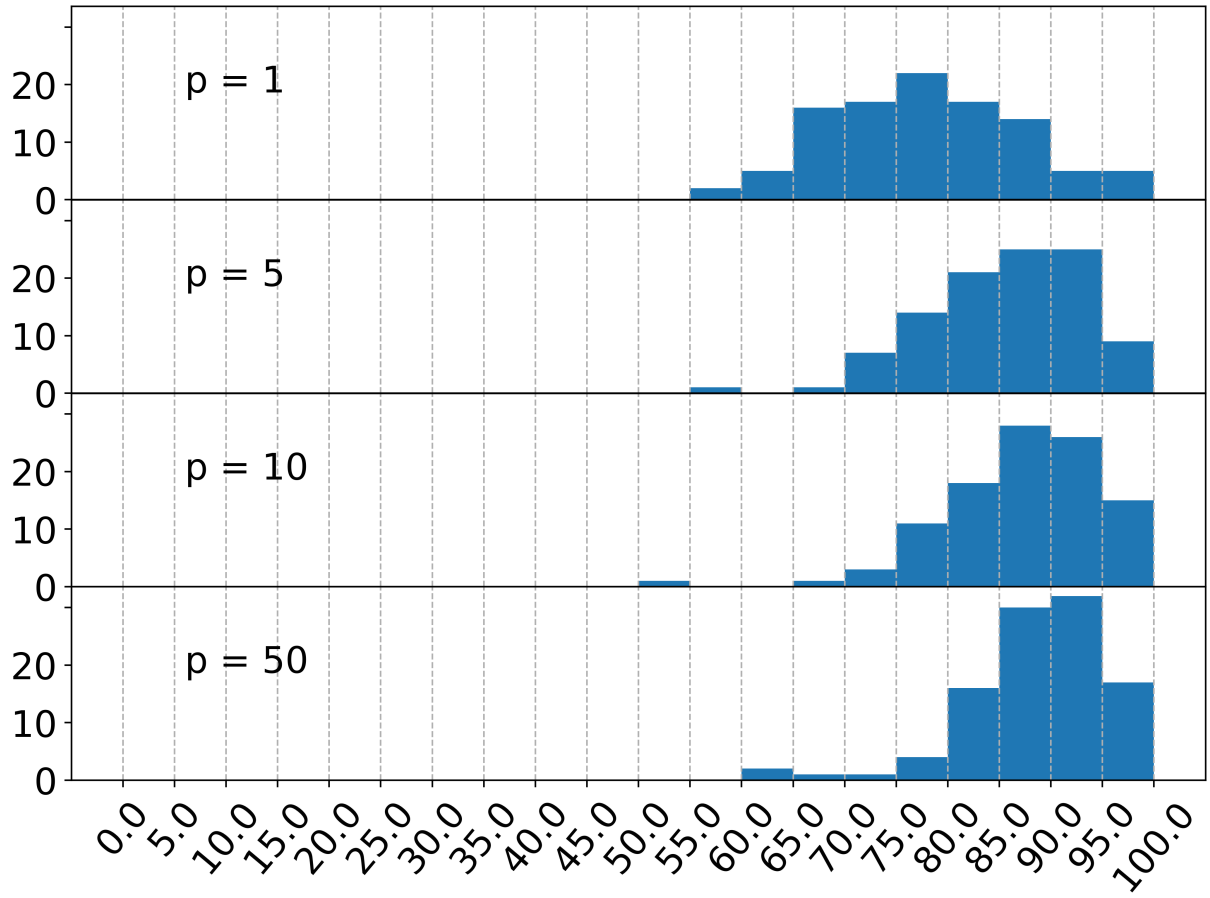

Figure 3.11: A histogram of the cross-validated accuracy of the deciding whether a message came from a Python group or an illicit one. Each row shows the dependence on the percentile of the data used.

	Accuracy						Matthew's coefficient					
	logreg		AdaBoost		ANN		logreg		AdaBoost		ANN	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
p = 1	79.84	8.11	80.07	7.67	82.36	8.01	0.49	0.19	0.5	0.18	0.56	0.19
p = 5	84.46	7.09	83.94	7.02	86.84	6.75	0.62	0.16	0.61	0.16	0.68	0.15
p = 10	86.1	6.71	84.93	6.82	87.95	6.23	0.66	0.15	0.63	0.15	0.71	0.14
p = 50	87.94	6.22	87.94	6.22	89.15	5.84	0.7	0.14	0.65	0.15	0.74	0.13

Table 3.1: The accuracy (in percentage) and Matthew's correlation coefficient for Logistic Regression, Adaboost and ANN, and percentiles of the data. The ANN is the best algorithm, but the difference is not large and within the observed variance. The difference between 10% and 50% is small compared to the increase in the work required.

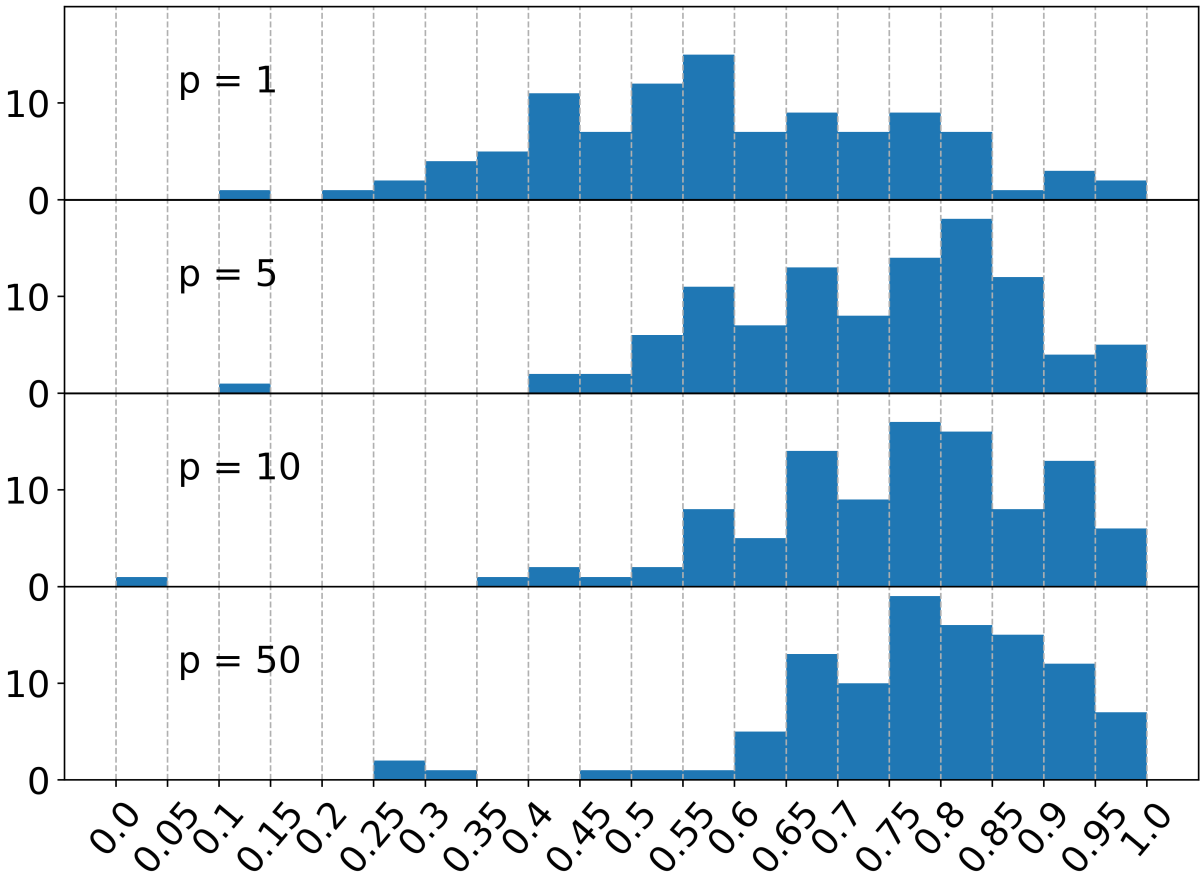

Figure 3.12: A histogram of the Matthew's correlation coefficient of the deciding whether a message came from a Python group or an illicit one. Each row shows the dependence on the percentile of the data used.

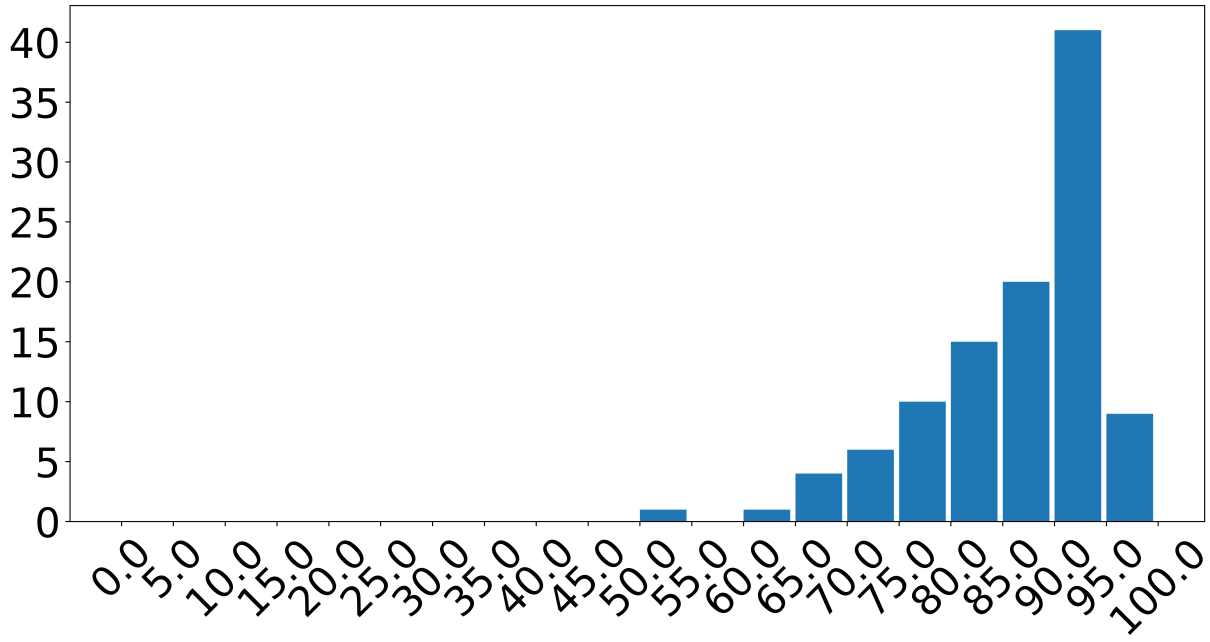

Figure 3.13: A histogram of the cross-validated accuracy of the deciding whether a message came from a Telegram group or Twitter. The 10% percentile was used for this figure.

time, which makes it eminently suitable for big data problems.

We successfully classified the originating group of 5,253 pairs of Telegram chats, distinguished bot messages from human ones, identified listings or advertisements, and identified human comprehensible differences between illicit conversations and Python programming conversations. Our approach also could identify the difference between social networking entities, distinguishing Twitter from Telegram. The approach is considerably more accurate than embedding algorithms like Doc2Vec on this kind of data. It is also much faster than Doc2Vec. It is a little less accurate than TF-IDF, but uses far fewer features.

3.6.2 Future Work

Future work includes extending the algorithm to handle phrases and sets of words. This should improve the accuracy while keeping the efficiency. However, it may require an extra unsupervised machine learning step to extract the most meaningful phrases much as is used in embedding algorithms. Combinations of this approach with embedding, at least at the word level, could be useful for larger and longer documents. We could also explore the use

of an “inverse document frequency”, the IDF in TF-IDF, to improve accuracy. Finally, the work should be extended to include non-literary features, such as emojis and non-ASCII UTF-8 and UTF-16 characters.

PART 4

CONCLUSION

As a conclusion, this dissertation follows process towards graph-information retrieval using feature engineering:

4.0.1 Defining the base problem as a binary classification

Given datasets, we first posed the questions in terms of binary classifications. In the case of HIV protease, we posed the question of binary classification: Given an HIV protease, could we tell if it is resistant or not? Given a resistant HIV protease, couuld we tell if the resistance level is high or low? In case of the telegram messages, we posed the question of binary classification: given a message, and two types of authors, could we tell which one is the most likely author of the message?

4.0.2 Understanding the information that holds value with the questions asked

Given the dataset and posed the questions, understanding the information that holds importance with respect to relations. In the case of HIV protease mutations dataset, the need of its representation being translational and rotational invariant and preservation of the relative amino acid structure arose from the data being a protein. The need of data being a metric that represents the evolution of the protein arose from that fact that the questions being asked were across the mutations representing different time, making the underlying graph a metric representation of entities across the time. In the case of telegram groups messages dataset, the training examples for messages pertaining to each type of author were already given. The need of finding the words that represented their special usage in the contrast of two given sets arose from the question that was being asked: determining the author of the message.

As illustrated in both the datasets, the understanding of the important information

is very domain specific. There are several ways to obtain this understanding, including representing the purpose of the underlying objects that are to be represented as vectors as in HIV protease dataset, or empirically observing the word usage that was special in the telegram dataset.

4.0.3 Mathematically translating this important information to sparse feature vectors

Once understood, the important information was translated into a mathematical representation. In the case of HIV protease, the Delaunay Triangulation, a model that satisfied the needs of translational and rotational invariance and preservation of the relative molecule structure was used to represent the HIV protease. In the case of telegram message, the words serving as idiomatic features were noticed to be captured by relative Kullback measure, and the messages were represented as sparse vectors with respect to the bag of these informed words.

Finding the existing models such as Delaunay Triangulation and relative Kullback measure that befitted the information that was crucial to answering the questions required the concrete understanding on what was needed from the information and was highly domain specific.

4.0.4 Graph construction on the nodes that are represented by the sparse feature vectors

The edges of the graph came from the underlying data,as structural similarity. In the HIV protease case, the nodes represented the HIV protease, and the edges represented the similarity between the HIV protease's vector representation. In the case of telegram data, the edges represented the messages, and the edges represented the structural similarity between message representations constructed with the same informed bag of words.

4.0.5 Validation of this new feature vectors as representation of the entities that they were supposed to represent

Once constructed, these feature vectors that were supposed to represent the HIV protease and telegram messages were tested with three fold cross validation to answer the binary classification questions posed at the beginning. We did so by using shallow classifiers such as SVM or ANN. The underlying assumption was that the vector similarity between two objects meant that these objects belonged to the same class. We could use these shallow classifiers because the vector representations were sparse, and captured the essential relations in the information that was required to make a high accuracy classification.

4.0.6 Finding the new information using the similarity through the sparse representation

Since the sparse representation of the underlying objects was found to be robust as it answered the classification questions with high accuracy, we could now use the same similarity to obtain new information using the existing graph information retrieval techniques. In the case of HIV protease, sketching the minimal spanning tree of these HIV protease told us the story of the evolution, since the similarity between the two HIV proease could be interpreted across the time. In the case of telegram messages, the similarity between two messages yielded a natural clustering of its authors and illustrated the characteristics of these authors in terms of their language usage patterns. Since the similarity between messages could be interpreted in terms of the kind of English used, the vectors of illicit messages also let us build a prototype of the "English" used by the criminals.

4.0.7 Lowering computational cost and still having comparable goodness metric

One of the main purposes of this dissertation is to device feature engineering to reduce computational cost without suffering an incomparable reduction in the accuracy, precision, recall, f1 score or Matthew's correlation coefficient. The results show that the sparse features obtained by the methods in this dissertation result in lowering the time and computational

complexity enough that the shallow classifiers such as SVM and ANN resulted in stellar goodness metric scores. The results generated by this dissertation did not have to be trained on special hardware such as CPUs or GPUs and used computationally expensive models such as deep neural networks, but could still answer the questions with high efficiency. The same questions could have not been answered without these special accommodations without the understanding of the domain knowledge and choosing the right mathematical models. The route without feature engineering when taken could be studied in [69] [70] where the results obtained heavily depend on the complex systems that reduce the apriori domain knowledge.

For the rest of the chapter, we list the conclusions of each of the studies separately.

4.1 Concluding the study of HIV protease and drug resistance

The section 2 shows the results in the two main aspects: The goodness of the feature-engineered sparse vector representations SWED and RSWED, and the new information gain by using SWED and RSWED similarities to track HIV protease across the time.

In particular, The sparse vector representation of HIV protease captures is very effective proxying the HIV protease genome sequence for tracking the evolution of HIV protease under the influence of most popular HIV protease inhibiting drugs. This is evident by the average accuracy of 99.5% of the binary classification of the drug resistance. This research also demonstrated that the same feature representation could be used for the regression purposes to know the drug resistance as a continuous number.

The SWED and RSWED vectors are generated simply by counting the neighbors of each Amino Acid of HIV protease in its Delaunay triangulation. This Delaunay triangulation is calculated just once on the base sequence of HIV protease and used multiple times for each HIV protease sequence that may have been a mutant of the base sequence. Thus, the vectors are normalized by the length of 210, this length being the non-repeating information of 20x20 adjacency matrix of the 20 types of Aamino Acids in HIV protease. Thus, the complexity of the vector generation is linear, and the vector generation could be executed on a single, non-expensive computer.

As a bi-product of being the neighborhood of a Delaunay triangulation, these vector representations of HIV protese are also sparse. These sparse and normalized vectors could use the shallow learning methods such as linear SVM with high effectiveness of average 99.5% accuracy by the virtue of sparsity. These shallow classifiers could be generated and trained on a single, non-expensive computer. Thus, the process of generating the feature-engineered vectors and training a shallow classifier is highly cost effective.

The answer to the question of "how does the resistance of HIV protease change with evolution" could be found by using the same vector representations of HIV protease by viewing the evolution as a graph across the time. The nodes of this graph are the vector representations of the HIV protease and edges are the distances between these vectors. The underlying assumption about this graph representing the evolution of HIV protease across the time is that the further a representation is from that of the base sequence, the further it mutated from the base and hence the further in time it is generated. To put these protease nodes in a feasible evolution time frame, a minimal spanning tree of this graph was generated. The root of the tree is perceived to be the base sequence of the HIV protease. The minimal spanning tree represents the shortest paths from root to leaves. In the case of the HIV protease, the minimal spanning tree thus represented a possible evolution mechanism from the base sequence to be taken as start of the evolution and each edge representing the next mutation. The study of the branches of this tree thus could yield the understanding of the evolution of the HIV protease with respect to the resistance. By studying the branches of this tree from roots to leaves, we found that the branches that converted to resistance early on in the evolution, not only remained resistant, but the resistance level increased with the time. We found a few branches that stayed non-resistant through the time, but the number of these branches was much lower. The branches that converted to being resistant later on could not preserve its resistance or were not highly resistant with the time.

This tendency of HIV to conserve resistance could be explained by HIV's high rate of mutation and basic principle of evolution: Highly resistant HIV's mutations are also likely to be resistant, and hence HIV reacts to the proteasae inhibitors through selective pressure

of being resistant.

4.2 Concluding the language and ownership of messages in telegram's illicit and licit groups

Similar to the HIV protease dataset, the section 3 shows the results in two main aspects: The goodness of the feature-engineered IBOW vectors in terms of sparsity, computational complexity, and classification accuracy, and the new information gain obtained by using this feature-engineered vectors.

In the setup of this research, we are given two sets of messages pertaining to two different entities and the question we ask is that given a new message, which set does it belong to. As discussed in the section 3, the current approaches maximize the apriory information by considering a union of the words in both given sets of messages. In contrast to these approaches, IBOW is generated by taking only 20% of the intersecting words of both the sets. IBOW vectors thus result in shorter and sparser vectors with respect to its comparable statistical and deep-learning counterparts. IBOW vector generation could be classified algorithmically as local search in contrast to global optimization approach of the other methods. IBOW identifies the words that are common in both sets but are used in different context and hence have highly imbalanced frequency. This is done by calculating relative Kullback measure of all the intersecting words. Hence the computational complexity of IBOW vector generation is quadratic in the number of intersecting words, which is achievable on a non-expensive compute node locally.

The primary objective of IBOW vectors is to represent the given messages that could be used to classify the authorship of the messages. IBOW vectors serve this purpose ideally because they sparcifies the text data while preserving the features that distinguish the ownership of the message. In addition, by the virtue of their sparsity, IBOW vectors can be used to train shallow learning models such as logistic regression, adaboost and artificial neural networks. Again, this shallow learning could be computed on a single, non-expensive compute node. These shallow learning classifiers trained on IBOW vectors result in the good-

ness scores that are comparable to computationally expensive and complex statistical and deep learning methods. Once trained, this classifiers could be used on smaller computers, including field deployable devices that are not much more powerful than a cell phone.

The information gain by using IBOW vectors is the grammar free understanding of the language that is used in the criminal domain the models were trained on. In particular, IBOW vectors theme-selves represent the list of words that are idiomatic features and hence are used in distinct contexts in the criminal lingo and vernacular English. Thus, this study not only results in powerful classifiers automating the criminal message recognition process, but also lead us to build a language prototype used as "English" in the criminal domain.

PART 5

APPENDIX

5.1 Works published

- Shah, Dhara, T. G. Harrison, Christopher B. Freas, David Maimon, and Robert W. Harrison. "Illicit Activity Detection in Large-Scale Dark and Opaque Web Social Networks." In 2020 IEEE International Conference on Big Data (Big Data), pp. 4341-4350. IEEE, 2020.

- Shah, Dhara, Christopher Freas, Irene T. Weber, and Robert W. Harrison. "Evolution of drug resistance in HIV protease." BMC bioinformatics 21, no. 18 (2020): 1-15.

- Abeysinghe, Bhashithe, Dhara Shah, Chris Freas, Robert Harrison, and Rajshekhar Sunderraman. "POSLAN: Disentangling Chat with Positional and Language encoded Post Embeddings." arXiv preprint arXiv:2107.03529 (2021).

- Freas, Christopher B., Dhara Shah, and Robert W. Harrison. "Accuracy and Generalization of Deep Learning Applied to Large Scale Attacks." In 2021 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE, 2021.

- Shah, Dhara, Yubao Wu, Sushil Prasad, and Danial Aghajarian. "Connected-Dense-Connected Subgraphs in Triple Networks." arXiv preprint arXiv:2011.09408 (2020).

- Shah, Dhara; Prasad, Sushil; and Aghajarian, Danial, "Finding densest subgraph in a bi-partite graph" (2019). Computer Science Technical Reports. 1.

- Shah, Dhara; Prasad, Sushil; and Wu, Yubao, "Finding Connected-Dense-Connected Subgraphs and variants is NP-Hard" (2019). Computer Science Technical Reports. 2.

- Khare, Alind, Vikram Goyal, Srikanth Baride, Sushil K. Prasad, Michael McDermott, and Shah, Dhara. "Distributed Algorithm for High-Utility Subgraph Pattern Mining Over Big Data Platforms." In 2017 IEEE 24th International Conference on High Performance Computing (HiPC), pp. 263-272. IEEE, 2017.

- Prasad, Sushil K and Aghajarian, Danial and McDermott, Michael and Shah, Dhara and Mokbel, Mohamed and Puri, Satish and Rey,Sergio and Shekhar, Shashi and Xe, Yiqun and Vatsavai, Ranga Raju and Wang, Fusheng and Liang, Yanhui and Vo, Hoang and Wang, Shaowen. Parallel Processing over Spatial-Temporal Datasets from Geo, Bio, Climate and Social Science Communities: A Research Roadmap. IEEE Big Data Congress, 2017.

- Prasad, Sushil K and Shekhar, Shashi and Zhou, Xun and McDermott, Michael and Puri, Satish and Shah, Dhara and Aghajarian, Danial. A Vision For GPU-accelerated Parallel Computation on Geo-Spatial Datasets. Sigspatial Newsletter Special issue on Big Spatial Data, 2014

5.2 Works in progress

- Shah, Dhara et. al Telegram, "lay of the land", financial fraud domain – In preparation

- Shah, Dhara, Christopher Freas, Irene T. Weber, and Robert W. Harrison. Visualizing evolution of drug resistance in HIV proterase – in preparation

REFERENCES

[1] J. R. Nurse and M. Bada, "The group element of cybercrime: Types, dynamics, and criminal operations," *arXiv preprint arXiv:1901.01914*, 2019.

[2] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[3] ——, "Studies on protein stability with t4 lysozyme," *Advances in protein chemistry*, vol. 46, pp. 249–278, 1995.

[4] W. H. Organization. World health organization hiv paget. Access: 7/31/2019. [Online]. Available: http://www.who.int/hiv/data/en/

[5] H. Wang, T. M. Wolock, A. Carter, G. Nguyen, H. H. Kyu, E. Gakidou, S. I. Hay, E. J. Mills, A. Trickey, W. Msemburi *et al.*, "Estimates of global, regional, and national incidence, prevalence, and mortality of hiv, 1980–2015: the global burden of disease study 2015," *The lancet HIV*, vol. 3, no. 8, pp. e361–e387, 2016.

[6] R. P. Smyth, M. P. Davenport, and J. Mak, "The origin of genetic diversity in hiv-1," *Virus Research*, vol. 169, no. 2, pp. 415 – 429, 2012, retroviral RNA, protein co-factors and chaperones. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168170212002122

[7] I. T. Weber and R. W. Harrison, "Decoding hiv resistance: from genotype to therapy," *Future Medicinal Chemistry*, vol. 9, no. 13, pp. 1529–1538, 2017, pMID: 28791894. [Online]. Available: https://doi.org/10.4155/fmc-2017-0048

[8] M. W. Chang and B. E. Torbett, "Accessory mutations maintain stability in drug-resistant hiv-1 protease," *Journal of Molecular Biology*, vol. 410, no. 4, pp.

756 – 760, 2011, structural and Molecular Biology of HIV. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022283611003147

[9] T. R. Weikl and B. Hemmateenejad, "Accessory mutations balance the marginal stability of the hiv-1 protease in drug resistance," *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 3, pp. 476–484, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25826

[10] X. Yu, I. Weber, and R. Harrison, "Sparse representation for hiv-1 protease drug resistance prediction," in *Proceedings of the 2013 SIAM international conference on data mining*. SIAM, 2013, pp. 342–349.

[11] X. Yu, I. T. Weber, and R. W. Harrison, "Prediction of hiv drug resistance from genotype with encoded three-dimensional protein structure," *BMC genomics*, vol. 15, no. 5, p. S1, 2014.

[12] ——, "Identifying representative drug resistant mutants of hiv," *BMC bioinformatics*, vol. 16, no. 17, p. S1, 2015.

[13] E. E. A. Durham, X. Yu, and R. W. Harrison, "FDT 2.0: Improving scalability of the fuzzy decision tree induction tool - integrating database storage," in *2014 IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE)*, Dec 2014, pp. 187–190.

[14] C. Shen, X. Yu, R. W. Harrison, and I. T. Weber, "Automated prediction of hiv drug resistance from genotype data," *BMC bioinformatics*, vol. 17, no. 8, p. 278, 2016.

[15] H. Tingjun, Z. Wei, W. Jian, and W. Wei, "Predicting drug resistance of the hiv-1 protease using molecular interaction energy components," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, no. 6, pp. 1099–1099, 2014.

[16] "Sheik Amamuddy, Olivier and Bishop, Nigel T. and Tastan Bishop, Özlem", "Improving fold resistance prediction of hiv-1 against protease and reverse

transcriptase inhibitors using artificial neural networks," *BMC Bioinformatics*, vol. 18, no. 1, p. 369, Aug 2017. [Online]. Available: https://doi.org/10.1186/s12859-017-1782-x

[17] M. Masso and I. I. Vaisman, "Sequence and structure based models of hiv-1 protease and reverse transcriptase drug resistance," *BMC genomics*, vol. 14, no. 4, p. S3, 2013.

[18] P. Bose, X. Yu, and R. W. Harrison, "Encoding protein structure with functions on graphs," in *2011 IEEE international conference on bioinformatics and biomedicine workshops (BIBMW)*. IEEE, 2011, pp. 338–344.

[19] L. Ramon, Elies Belanche-Muñoz and M. Pérez-Enciso, "Hiv drug resistance prediction with weighted categorical kernel functions," *BMC Bioinformatics*, vol. 20, no. 1, p. 410, 2019.

[20] S. D. Pawar, C. Freas, I. T. Weber, and R. W. Harrison, "Analysis of drug resistance in hiv protease," *BMC bioinformatics*, vol. 19, no. 11, p. 362, 2018.

[21] W. M. Fitch, "Phylogenies constrained by the crossover process as illustrated by human hemoglobins and a thirteen-cycle, eleven-amino-acid repeat in human apolipoprotein a-i," *Genetics*, vol. 86, no. 3, pp. 623–644, 1977. [Online]. Available: https://www.genetics.org/content/86/3/623

[22] G. J. Szöllősi, E. Tannier, V. Daubin, and B. Boussau, "The Inference of Gene Trees with Species Trees," *Systematic Biology*, vol. 64, no. 1, pp. e42–e62, 07 2014. [Online]. Available: https://doi.org/10.1093/sysbio/syu048

[23] M. D. Rasmussen and M. Kellis, "A Bayesian Approach for Fast and Accurate Gene Tree Reconstruction," *Molecular Biology and Evolution*, vol. 28, no. 1, pp. 273–290, 07 2010. [Online]. Available: https://doi.org/10.1093/molbev/msq189

[24] R. R. Hudson, M. Slatkin, and W. P. Maddison, "Estimation of levels of gene flow from dna sequence data." *Genetics*, vol. 132, no. 2, pp. 583–589, 1992. [Online]. Available: https://www.genetics.org/content/132/2/583

[25] G. Bello, C. Casado, S. García, C. Rodríguez, J. del Romero, and C. López-Galíndez, "Co-existence of recent and ancestral nucleotide sequences in viral quasispecies of human immunodeficiency virus type 1 patients," *Journal of general virology*, vol. 85, no. 2, pp. 399–407, 2004.

[26] I. T. Weber, D. W. Kneller, and A. Wong-Sam, "Highly resistant hiv-1 proteases and strategies for their inhibition," *Future Medicinal Chemistry*, vol. 7, 2015.

[27] S.-Y. Rhee, M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database," *Nucleic acids research*, vol. 31, no. 1, pp. 298–303, 2003.

[28] Filtered phenosense data. Accessed: 7/15/2019. [Online]. Available: https://hivdb.stanford.edu/download/GenoPhenoDatasets/PI_DataSet.txt

[29] R. F. Ling, "On the theory and construction of k-clusters," *The computer journal*, vol. 15, no. 4, pp. 326–332, 1972.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[31] Evolution of drug resistance in hiv protease. Accessed: 9/1/2019. [Online]. Available: https://github.com/hithisisdhara/HIV_protease

[32] Y. Tie, A. Y. Kovalevsky, P. Boross, Y.-F. Wang, A. K. Ghosh, J. Tozser, R. W. Harrison, and I. T. Weber, "Atomic resolution crystal structures of hiv-1 protease and mutants v82a and i84v with saquinavir," *Proteins: Structure, Function, and Bioinformatics*, vol. 67, no. 1, pp. 232–242, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21304

[33] A. Hagberg, D. Schult, P. Swart, D. Conway, L. Séguin-Charbonneau, C. Ellison, B. Edwards, and J. Torrents, "Networkx," *URL http://networkx. github. io/index. html*, 2013.

[34] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *Third international AAAI conference on weblogs and social media*, 2009.

[35] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PloS one*, vol. 9, no. 6, 2014.

[36] D. Maimon, Y. Wu, N. Stubler, and P. Sinigirikonda, "Extended validation in the dark web: Evidence from investigation of the certification services and products sold on darknet markets," 2020.

[37] F. McCown and M. L. Nelson, "A framework for describing web repositories," in *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 341–344. [Online]. Available: https://doi.org/10.1145/1555400.1555456

[38] P. Wang, X. Liao, Y. Qin, and X. Wang, "Into the deep web: Understanding e-commerce fraud from autonomous chat with cybercriminals," in *NDSS*, 2020.

[39] R. A. HARDY and J. R. NORGAARD, "Reputation in the internet black market: an empirical and theoretical analysis of the deep web," *Journal of Institutional Economics*, vol. 12, no. 3, p. 515–539, 2016.

[40] O. Babko-Malaya, R. Cathey, S. Hinton, D. Maimon, and T. Gladkova, "Detection of hacking behaviors and communication patterns on social media," in *2017 IEEE international conference on big data (Big Data)*. IEEE, 2017, pp. 4636–4641.

[41] W. Sayers, "Cant, rant, gibberish, and jargon," *ANQ: A Quarterly Journal of Short Articles, Notes and Reviews*, vol. 28, no. 1, pp. 1–10, 2015.

[42] J. P. Considine, *Small dictionaries and curiosity: Lexicography and fieldwork in post-medieval Europe*. Oxford University Press, 2017.

[43] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[44] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016, pp. 78–86.

[45] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[46] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Fifth International AAAI conference on weblogs and social media*. Citeseer, 2011.

[47] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the workshop on language in social media (LSM 2011)*, 2011, pp. 30–38.

[48] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, 2018.

[49] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, 2009.

[50] S. Ghosh, A. Das, P. Porras, V. Yegneswaran, and A. Gehani, "Automated categorization of onion sites for analyzing the darkweb ecosystem," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1793–1802.

[51] N. Tavabi, N. Bartley, A. Abeliuk, S. Soni, E. Ferrara, and K. Lerman, "Characterizing activity on the deep and dark web," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 206–213.

[52] R. Bhalerao, M. Aliapoulios, I. Shumailov, S. Afroz, and D. McCoy, "Mapping the underground: Supervised discovery of cybercrime supply chains," in *2019 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2019, pp. 1–16.

[53] J. Li, Q. Xu, N. Shah, and T. K. Mackey, "A machine learning approach for the detection and characterization of illicit drug dealers on instagram: model evaluation study," *Journal of medical Internet research*, vol. 21, no. 6, p. e13803, 2019.

[54] "Telegram groups: A list of 350+ groups in english," https://telegramchannels.me/groups, (Accessed on 08/18/2020).

[55] "1000+ best telegram group link 2020 (search to join a chat)," https://telegramguide.com/telegram-group-link/, (Accessed on 08/18/2020).

[56] "8000+ telegram channels, groups, bots and stickers list," https://telegramchannels.me/, (Accessed on 08/18/2020).

[57] "Telegram group: Find telegram channels, bots & groups," https://www.telegram-group.com/en/, (Accessed on 08/18/2020).

[58] "Top telegram groups," https://combot.org/telegram/top/groups, (Accessed on 08/18/2020).

[59] "United states telegram group link search — us tg list," https://www.hottg.com/telegram-group/us, (Accessed on 08/18/2020).

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

[61] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[62] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, Jun. 2008. [Online]. Available: https://doi.org/10.1145/1361684.1361686

[63] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, 2004.

[64] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

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

[66] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, vol. 1.

[67] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, p. 22, 2011.

[68] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, p. 6, 2020.

[69] N. Kaur and W. Ghai, "Performance analysis of deep cnn assisted optimized hiv-i protease cleavage site prediction with hybridized," in *International Conference on Communication, Computing and Electronics Systems*. Springer, p. 529.

[70] B. Abeysinghe, D. Shah, C. Freas, R. Harrison, and R. Sunderraman, "Poslan: Disentangling chat with positional and language encoded post embeddings," *arXiv preprint arXiv:2107.03529*, 2021.