Cyber Persona Identification via Indirect Feature Analysis

Suzanne Stathatos[1], Asitang Mishra[1], Chris A. Mattmann[1,2]
chris.a.mattmann@jpl.nasa.gov

[1]Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109 USA

[2]Computer Science Department
University of Southern California
Los Angeles, CA 90089 USA

ABSTRACT

In this paper, we present an approach to identify users across website forums using indirect features derived from metadata on each of the websites. The approach combines graph analysis techniques with orthographic and phonological analyses, and with extracting raw metadata from each user on the websites. Using fourteen features derived from these techniques, all pairs of users from the two separate websites were created. Pairwise features for each of the user pairs then determined the similarity scores between users across the website forums. Our technique is both scalable to thousands of users, and accurate achieving an F-score of 0.963. when evaluated on a dataset crawled from dark market forums from the DARPA MEMEX program.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Applied computing** → *Computer forensics*;

KEYWORDS

Persona Extraction, Author identification, Feature Identification, Information Retrieval

ACM Reference Format:
Suzanne Stathatos[1], Asitang Mishra[1], Chris A. Mattmann[1,2]. 2018. Cyber Persona Identification via Indirect Feature Analysis. In *Proceedings of ACM Workshop on Graph Techniques for Adversarial Activity Analytics (GTAş 2018).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Today's web is home to abundant e-markets and sales-oriented websites including Amazon, eBay and other properties whose sales are approaching trillions of dollars[1]. Along with legitimate transactions, the web is also home to many illegal and illegitimate transactions as well. Forum sites in which users illegal trade in automatic weapons [6], dark markets where illicit pharmaceuticals are traded for anonymous currency; public forums where humans are trafficked [16], are examples of such nefarious activity, to name a few.

[1]http://fortune.com/2017/03/31/amazon-stock-trillion-dollar-company-apple-tesla-google/

GTAş 2018, February 2018, Marina Del Rey, CA USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

Identifying and tracking user activity across these forums is a key capability for law enforcement. Criminals typically sell their wares across multiple sites, using different registered users, and in geographically disparate areas. Sometimes the seller of illegal products can be a person; other times a group, business, or a government - we define these entities as *cyber persona*s. The general area of cyber persona detection aims to discover the geography, function, and attributes of the target persona. While the physical plane (e.g. hardware, operating characteristics, physical environment) and the logical plane (e.g. OS, configuration, IP address) offer insight into geographic location and persona attributes, techniques are needed to focus on the contextual plane (ownership, affiliation, user, language, timezone) in a 1-to-many or many-to-many relationship.

Cyber persona attributes can include identifying information – local and remote user accounts, services used (e.g. ICQ, email, web browser, Tor), certificates, user identification, images, audio, bitcoin, etc. – and personal information including associates, employment, education, address, sex, height, age, weight, marital status, family, hobbies and other demographic information. Constructed personas may reveal personally identifiable information(PII) or may be purposely deceptive. Because trust is a desired outcome, the process of establishing reputation may also reveal useful information. Developing personas are important to the "user", so many real-world attributes of the "user" are manifested in the persona (e.g. sense of self, purpose).

Connecting user identities across website forums poses interesting challenges. The websites' users may or may not overlap. For example, $website_1$ may have users A, B, and C and $website_2$ may have users D, F, and A. In this example, only user A is on both websites. Furthermore, the websites' metadata may differ. For example, $website_1$ may maintain username, posts, post time, Facebook handle, Twitter handle, register time, and user type, whereas $website_2$ may have username, posts, post time, phone number, email address, and profile picture. While both websites combined have several components, they only have a few matching metadata components. In addition, the matching metadata components may differ, as the user may not be posting at the exact same time on both sites, the user may not post the exact same content, and the user may be more active on one site as compared with another, to name a few differences. Regardless of these differences, there still exists a mapping between the same users across websites.

During the last three years, our research group has completed efforts working with The Defense Advanced Research Projects Agency (DARPA) and its Memex program in a variety of areas including cyber persona detection. Our technique in particular is well suited to the domain as we leverage indirect *metadata* features

(e.g., post times, causal relationship between posts; post text, post location, etc.) rather than hard links (e.g., usernames, PII, accounts, services, etc.) to establish persona profiles, and then to develop metrics of similarity between these profiles across a variety of forum websites using machine learning techniques including decision trees and random forest. Our technique is scalable across 1000s of users, and is accurate, achieving an F-score of 0.963 in our real-world datasets culled from dark market websites and deep web crawled forums.

Section 2 describes Background and Related Work in the area. Section 3 describes the indirect features we derived from our dataset. Section 4 describes how we evaluated feature accuracy and Section 5 describes the machine learning classifiers we built to identify personas across web forums. Section 6 describes the future work on our project, and Section 7 concludes the paper.

2 BACKGROUND AND RELATED WORK

Identifying the authors of web posts is similar to the task of username de-aliasing or profile matching. Some noteworthy work in this former that are used by the latter are [1, 19, 27]. In Pennebaker et al [19], the authors describe a software that extracts many psycholinguistic and stylometric features from text and is used by many researchers in the field of authorship detection. Abbasi et al [1] used lexical, syntactic, structural, content specific features based on the framework described by Zheng et al [27]. They also did principle component analysis (PCA) on their feature groups and then used sliding window algorithms to create visualizations for those features called writeprints, which can be compared visually to identify digital fingerprints of authorship.

In the areas of alias detection and user profile matching researchers have tried many different approaches. Most of the approaches involve some form of feature engineering like [9, 10, 20]. Johansson et al. [9] used four types of features. (1) String based features for username matching using jaro-winkler [25]; (2) Stylometric features e.g., word length, sentence length etc; (3) Time based, where they divided time of the day into windows and created time vectors and then took Euclidean distance between the time vectors; and (4) Social features based on graphs. Kaati et al. [10] and their approach (also used by Johansson et al.'s work) further investigates only posting time based features by creating various new ones. All the features they used have a discretized window based approach as their previous paper. Martijn et al. [20] along with using the same time features and n-grams also used more involved stylometric features compared to Johansson et al. They also tried using the new posting time based features from Kaati et al. but did not get better results for their dataset.

A large body of research considers the problem of matching profiles from different sites by creating profile vectors. Researchers try to do an end to end analysis from finding good features to classifying new pairs and even reducing the amount of time to calculate all the pairs [13, 14, 22]. Vosecky et al. [22] provide an unsupervised method to do profile matching. They create profile vectors for each user across multiple forums and then give a weight based comparison technique to score the similarity between two vectors. They use a threshold to decide if a profile is the same of not. Vosecky et al. also use three kinds of matchings: exact, partial and fuzzy to match different fields in their vectors. For partial and fuzzy matchings they describe namely VMA and VNM algorithms respectively, which are very similar to meta levenshtein that we have used in our experiments and that will be described in the next section. Qiang et al. [13] use a similar three pronged matching technique and also investigated reduction of the amount of time for pairwise matching. They called it perfect, quasy-perfect and partial match. Unlike the previous paper they used all the matchings to all the fields in a weighted manner and then applied weights to each kind of match which they learned from regression. Qiang et al. created two kinds of tokens from different profile fields namely exact tokens and derived tokens. They also indexed the tokens to avoid matching profiles that do not have any tokens in common. Malhotra et al. [14] use a supervised profile vector based approach. They used supervised classification of profile vectors including the profile image. Some papers focussed more on the string matching side of things including usernames and names of friends, etc. [2, 8, 12].The main analysis in [12] concerns usernames. The authors create a labelled training dataset in an unsupervised manner based on n-gram probability of usernames based on [23] and [24] and used classification via profile metadata, social relationship and post content based features. Juan et al. [2] use string features from the profiles with a number of different string matching algorithms. The authors showed the comparison of results using different string matching techniques. Finally Laikhram et al. [8] leverage string matching of usernames and social information including a user's "likes" and "follows" to find matches of a set of users from Facebook to Twitter.

3 DATASETS AND FEATURES

On the DARPA MEMEX project we were able to leverage scraped web data from a number of dark markets and forums. Due to the sensitive nature of the data, we will only describe its general characteristics. The data included scraped user information from four forums, which we will call forum 17 and forum 14 – we use these later as our testing dataset. The other two forums, ForumA and ForumB were used for our training dataset - we will describe this process later in Section 4.2. This scraped information for testing includes 729 users on one website (forum 17) and 726 users on a different website (forum 14). We generated a unique set of features indicating user behavior, user vocabulary, user public web presence, and user interactivity. We generated the majority of these from relationships between cleaned json files that contained raw metadata representing the user profile information from the forum sites. Though, one feature was generated by querying public on-line presence as we will describe later. We processed the metadata and the external links into a set of individual features per user. From these individual features, we created pairwise features for each user pair across website forums. We have $729x726 = 529,254$ possible pair combinations, and we have 14 metrics. However, because some users had sparse post-information matrices (some users existed, but never posted), we downsampled to focus on only active users on each of the forums.

The users' individual features (IF) we used are: (IF1) *Username*, (IF2) *Vocabulary size*, (IF3) *Wordlist of User vocabulary*, (IF4) *Graph degree: in- and out-degree of user in graphically-expressed website*

forum, (IF5) *Reply count : number of times user replied to a post*, (IF6) *Average reply time : average time it takes a user to respond to a post*, (IF7) *Post frequency : average speed at which user posted to a website*, (IF8) *Post times, in days and minutes*, (IF9) *Filenames of images associated with and used by the user*, (IF10) *MD5 hashes of images used by the user*, (IF11) *List of subjects posted by the user*, (IF12) *Filenames of attachments attached to user posts*, (IF13) *Time(s) of registration on the website*, and (IF14) *Top 5 search results*.

The pairwise features (PF) derived from these features are: (PF1) *Meta-Levenshtein (edit distance) between usernames*, (PF2) *Size of vocabulary overlap (same vocabulary words)*, (PF3) *Euclidean distance in numeric vocabulary size*, (PF4) *Jaro Winkler post similarity*, (PF5) *In and out-degreeness difference*, (PF6) *Reply count overlap*, (PF7) *Difference in average post times*, (PF8) *Difference in post frequency*, (PF9) *Gaussian difference in post time patterns*, (PF10) *Image name overlap*, (PF11) *MD5 overlap*, (PF12) *Attachment filenames overlap*, (PF13) *Registration time difference*, and (PF14) *Size of set overlap of external links from username query*.

Our features split into five broad categories: (C1) Features indicative of a user's vocabulary and stylistic tendencies; (C2) Features from graphs of the website forum to indicate user activity in the context of the group; (C3) Features that represent change over time of these metrics; (C4) Logistical features; and (C5) Features indicative of a public presence. We will categorize the features along the five categories throughout the remainder of this section.

We define success in matching users by comparing the identified matches by our method with a given set of known user pairs. These known user pairs were provided by the same organization that cleaned the original data and had access to the original website forums and that provided the data to the DARPA MEMEX program. We classify user matches on a binary scale - either two users were the same or not.

3.1 C1: Features

We generated several features that relate to a user's vocabulary and writing style. Using information from post bodies on the forums and post subjects, as well as simple username, we generated the following. **Usernames** - We kept track of each user's username. The pair feature derived from usernames across website forums was the meta-levenshtein distance of the two usernames. The meta-levenshtein algorithm was based on the paper [18] which improves upon the soft-tfidf algorithm introduced by [4]. **Vocabulary Size** - A numeric count of how many unique words a user used throughout the forum. A word was defined as any collection of letters separated on each side by a space (from another word). We did not load and check a dictionary for if the word existed or not, and we did not try to correct misspelled words. Hypothesizing that the user may misspell or otherwise use the same non-words in other website forums, we chose to keep all words. The vocabulary size is simply a numerical count of how many unique words a user used in all of their posts and post subjects. The pairwise feature derived from the users' vocabulary sizes was the arithmetic difference in the pairwise vocabulary sizes. **Wordlist of User Vocabulary** - A list of unique words used by the user in a website forum. A word is the same word as defined above. We chose to go with unique words rather than keeping a mapping between word usage and word count to begin with a simpler problem. Though, keeping track of words and word count per word could provide additional insight. The pairwise feature derived from two users' word lists was the size of the set of overlapping words. If two users' words have no overlap, the set is null, so the size of the set is 0. **Post subjects** - Rather than breaking apart the words into an unordered list, this feature kept the words in their original order. It keeps track of post subjects used by each user. Post subjects are then compared as full sentences to see if users across websites used the same subject. We hypothesized that the same user may use the same post subject, verbatim. The pairwise feature from two users' post subjects list are the size of the set of matching post subjects. We hypothesize that users with mirrored post subjects are related or similar in some way. **Post similarity using N-Grams and Stylistic Features** - We used psycho-linguistic features inspired by the thesis [26] along with stylistic features, unigrams, bigrams and stemmed words to create our final bag of words. The complete list of all bag of word features that we used:

- bigrams, stemmed words, unigrams
- stylistic features: count of numerical words, count of punctuations, count of big words (>6), average, words per sentence, count of sentences, count of words
- psycholinguistic features: Coordinating conjunctions, proper nouns, happy words, sad words, first person pronouns second and third person pronouns, indefinite pronouns, quantifier words, tentative words (likely, possibly etc.), insight words (understand, realize etc.)

For each class of psycholinguistic features (sad, happy etc.) we added to the bag of words a signature unique to that class. To calculate the similarity we took the cosine similarity between the tf-idf of the bag of words.

3.2 C2: Features

For each forum, we generated graphs to represent and extract quantitative measures how users interacted with each other on the forums. The idea of modeling a web forum as a graph is natural, as it mimics the network structure of users' interactions. We first discuss the terminology relevant to each of the graphs followed by an explanation of how we use them as features for each of the users. **Graph terminology for time-based graph** - In the time-based graph, we graphically represented users' post time tendencies, and time-based activity on the forum. We speculated that an active user on $forum_{17}$ would be comparably active on $forum_{14}$. Further descriptions of all of the nodes (graph vertices) and relationships (graph edges) are described below. A *node* is a unique user on the website forum. Each node contains particular metadata, including: a user's username, a user's latest post time, and a list of the posts' post-identifiers. An *edge* goes from one node v to a node w if node v's user wrote a post immediately before node w. The latest post time attribute for each of the nodes helps determine the parent-child timewise relationship. So, for example, if user A posts at 10:01am, and user B posts at 10:02am, and there is no activity from any other user on the forum between user A's post and user B's post, an *edge* will exist between user A and user B. Mimicking time ordering, the edge will go from user A to user B. If the same user posts immediately after him or herself, an edge, or *cycle*, is

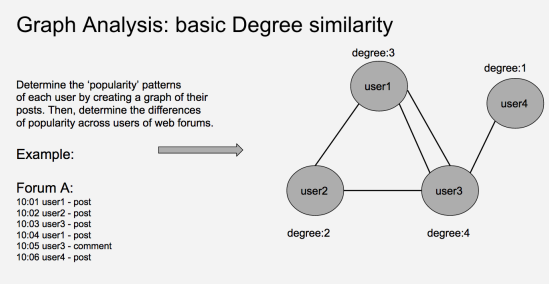

Figure 1: Degreeness of users in graph created from post time sequences.

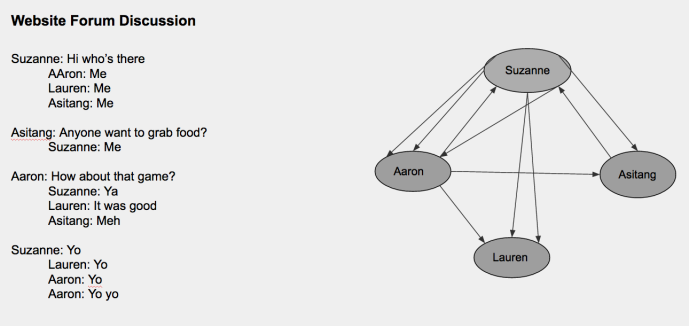

Figure 2: Forum turned to reply graph.

drawn from that user node back to itself. For example, if user A posts immediately after user A, an edge will be drawn from user A to user A. Figure 1 demonstrates the time-sequence graph of users.

In graph theory, the *degree* of a vertex is the number of edges incident to the vertex. The degree of a vertex, in other words, is the sum of *ingress* (going to the node) and *egress* (going away from the node) edges that touch the node. Loops, therefore, count twice. An *island node* is a vertex with degree 0. Island nodes indicate that the user has signed up for the website forum, but has never posted on it. Only two vertices could have degree 1 – a vertex that represents the user whose post began the forum and never continued with it, and a vertex that represents the user whose only post was the very last post on the forum. All other vertices will have a higher-valued degree. We call a user's *degreeness* the degree value of their vertex in this graph. The degreeness pairwise feature was the arithmetic difference between two users' degrees. **Graph terminology for threads-based graph** - In the thread-based graph, we aimed to capture direct interactivity among users. In this graph, a *node* is a unique user on a website forum. Each node contains a list of post ids for posts that that user had made. An *edge* in this graph indicates a reply to a post. If a directed edge exists from user A to user B, that indicates that user A posted something, and user B posted something as a reply to user A's original post. The direction of the edges matters in this graph because they differentiate the user that posted the parent post and the user that replied to that parent post.

Rather than rely on time sequences to indicate interaction, we leveraged the parent-post to child-post relationships annotated in the DARPA-provided dataset. The dataset, which appears to have been a snapshot of an online forum, contained post information. Each post was annotated with a post id and a parent post id. If post A's parent post id matched post B's post id, post B was the original post, and post A was a reply to post B. We used these data, then, to construct the graph. Because each user node contained a list of post ids, corresponding users were found via their post ids. Because that dataset was a snapshot, not all parent post IDs were found. If a reply had a parent post ID that was not included or otherwise found in the data provided, an edge would not exist between the parent node and the child node. Otherwise, an edge would exist between the parent, a user whose post id list contained the parent post id, and the child. Figure 2 depicts these website forum interactions (post and comments) translated to graph format.

We derived two features from this graph: average post time, which will be described in a later section, and reply count. **Reply count** - is a count of the ingress edges on nodes. Looking at Figure 2 again, Suzanne has a reply count of 2, Lauren has reply count 3, Aaron has reply count 3, and Asitang has reply count 2. The pairwise reply count feature was the arithmetic difference of two users' reply counts. **Social Circles** - We leveraged the thread-oriented graph of replies with the strongly indicative matching-username feature to find social circles within the forum. We hypothesized that, if two forums contained a quorum of overlapping users, cliques, or social circles, may form within the graph. We define a social circle as a set of users who talk amongst each other. A user in a forum may partake in multiple social circles, or the user may not be part of any social circles. This feature has also been used by others including [3] and [5]. Figure 3 illustrates an example of a social circle. It demonstrates how the reply-graph can be used with highly similar usernames to find other matching users within the same network.

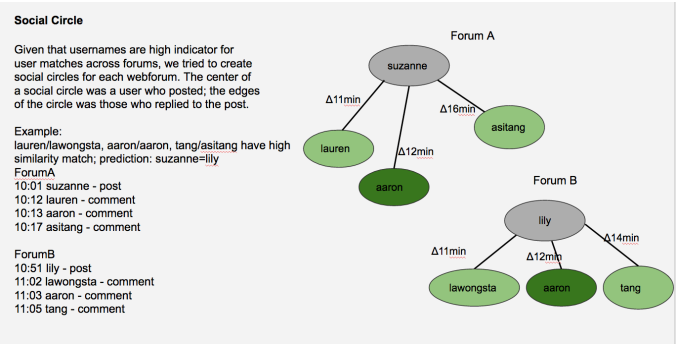

Figure 3: Social circles derived from reply graph.

3.3 C3: Features

We tried to capture time sequence properties within the graph, too, to demonstrate and analyze users' time-based behavior across website forums. We did this through three features: average post time, post frequency, and a bell curve fit for post activity during time intervals. These three features are described in further detail in this section. **Average Post Time** - The **average post time** is

the average amount of time it takes a user to post (with either a post or a comment in a thread) after the most recently issued post. The average time is illustrated in Figure 4. The average post time pairwise feature is the arithmetic difference in average post times.

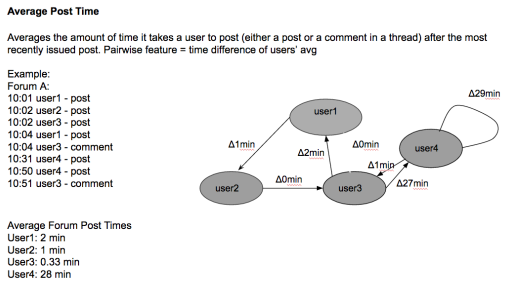

Figure 4: Depiction of how average post time is derived.

The hypothesis is that a user will have the same posting or reply tendency on one forum as (s)he has on another forum. So, if a user is more inclined to respond or post soon after another user, the same user probably has that tendency on the other forum as well. **Post frequency** - A user's **post frequency** is the the number of posts a user made divided by the sum of the post time differentials. This is essentially the inverse of the average post time. We included the frequency to see if it would include any new information lost in averaging the time differentials array. The post frequency pairwise feature was the arithmetic difference of users' post frequencies. **Post time habits through Gaussian extraction** - We extracted two kinds of feature from the timestamp of posts. To model the activity pattern of a user during a day as well as a week. To model the first , we the calculated mean and variance of the times during a day the used was active. We did the same for the second feature with days of the week. Then, we created a bell curve with the mean and variance and calculated the overlapping coefficient i.e. the intersection of area under the curve between two users to find a similarity measure we called Guassian Similarity.

3.4 C4: Logistical Features

Filenames of images associated with and used by the user - We maintained a list of the image filenames of images associated with the user, i.e. profile picture. The filenames pairwise feature was derived by intersecting the sets of filenames and counting how many filenames were found in the overlap. **MD5 image hashes** - We maintained a list of the MD5 hashes associated with the images used by the user. The MD5 pairwise feature was derived similarly to the filenames pairwise feature – we counted the number of MD5 in the intersection of the two sets. **Attachments' filenames** - Some users attached files to their posts. While the files themselves were not provided with the dataset, the file names were included if a file was attached to a particular post. We maintained a list of filenames that users used in their posts. The attachment filenames pairwise feature was derived by intersecting the sets of filenames and counting how many unique entities were in the intersection. **Registration times** - We kept note of when a user registered on a website forum, and if they had to register multiple times (possibly due to a complication with user type, if they forgot their password, etc.). If the user had multiple registration times, we summed the registration times together. Presumably, if a user had to register more than once for one website forum, the user may have to do the same on another forum. A sum of registration times would reflect multiple registration times. It would indicate relatively how many times a user had to register because the data from the forum does not span decades [11]. The registration time(s) pairwise feature was the arithmetic difference between the (potentially summed) registration times of the users.

User A	User B	Lev. Score	Meta-Lev Score
abacus	abacurse,	0.75	0.75
abacus1cat.	cat1cus	0.3	0.83

Table 1: Results from random forest classifier using username edit distances from sparse dataset.

3.5 C5: Features Indicative of a public presence

Google Queries by Username - We conducted a Google search on each of the usernames, and saved the top 5 resulting urls in a list. We postulated that if two users matched across forums, they might have similar, or the same, search results from the public web. The external links pairwise feature was the size of the intersection of the top 5 urls for users across website forums. The more links overlapped, presumably, the more likely the users were to be the same.

4 VERIFYING FEATURE ACCURACY

We began by proving that we could accurately find user matches across forums with a small subset of the features, namely, (1) usernames' meta-levenshtein distance, (2) user's post similarity using n-grams and stylistic features, and (3) overlapped posting patterns measured by timestamp bell curves. We trained a simple decision tree and random forest classifier using Python's `SciKitLearn` framework, and based on the provided known user pairs mentioned in the prior section - we will describe our classifiers more in Section 5. The python scripts that we used to parse and analyze the data are available at [17] and the scripts to create the individual user features and the distance features can be found at [15]. We applied these features to a more sparse dataset, with limited metadata. This data did not include thread information (making the construction of the reply graph impossible) or registration times. We considered only users that had post data. We filtered 374 pairs that had very similar usernames, and ran timestamp and post similarity feature extraction on roughly 23,000 pairs. Three hundred eighteen of the 1347 provided ground truth pairs had post data. Of the 318 pairs evaluated, 241 were reported correctly. Thus, from a smaller sample of users, and using fewer features, we found that roughly 3/4 of the users were matched successfully and 1/4 of the users were incorrectly unmatched. We also found username similarity to be the most important feature, as can be seen in Table 1, and activity during the day (timestamp bell curves) was the second most important feature. We harnessed this knowledge when expanding our feature set.

4.1 Data preparation for classification with full dataset and graph

Because so many of our features were influenced by users' posts, we included only samples that had post data. When classifying user pairs, we removed username-username labels, instead using only metalevenshtein scores to indicate name similarity. We made the post times graph and included the search link overlap features.

4.2 First Round - Training and Testing independence

We were provided with datasets from 4 different website forums. We will call the training forums ForumA and ForumB. We will call the testing forums Forum14 and Forum17. Initially, we used forums A and B to train our model, and we tested our model against Forum14 and Forum17. ForumA and ForumB did not contain metadata on parent post IDs, so our training and testing models included all features apart from the reply-graph-reliant features. In addition, ForumA and ForumB did not contain multiple registration times, so we instead took the first registration time from all users across all forums.

Because we were time-limited, rather than generating all possible pair combinations for these training and testing sets, we took all known true user matches, and randomly sampled three times the number of positives from the negative set (without replacement). We created a trained model from ForumA and ForumB. Then, put Forum14 and Forum17 through the same pipeline and tested against the trained model. We had 0 false matches, 6 true user matches, 14 user matches that were categorized as unmatched (false negatives), and the rest were true negatives, resulting in an F-score of 0.51.

Interested in how an entirely separate training dataset affected the results, we next used splits of Forum14 and Forum17 data as training and testing sets.

4.3 Second Round - Training and Testing splits

Feature adjustment - Using splits of the training and testing data from Forum14 and Forum17, we were able to incorporate our reply-graph-based features. However, we did not use the graph social circles, as too many users were connected with parent posts that were not provided in the dataset. To account for users that were incorrectly labeled as non-pairs, we included all users, including those that had never posted. Similar to before, due to time and computing constraints, we took a similar approach as before: we included all true user matches, and randomly sampled (without replacement) nine times that amount for negative matches. Because we knew all users in this training and testing set were able to have multiple registration times, we summed multiple registration times for this experiment. Each of the forums had easy, medium, and hard users to classify. Before splitting the forums into training and testing sets, we split them into their easy, medium, and hard sets. Once this was done, we split the data into 10-fold splits, making sure that a sufficient positive set of pairs was in each split. Thus, we trained on 90% of the data in each easy, medium, and hard class and tested on 10% of the data.

4.4 Feature importances

On the easy and medium datasets, the most important feature was the meta-levenshtein distance on usernames. This was largely became many of the same users across the easy and medium datasets used the same username. This was not the case in the hard data set. The hard data set classified the following as the most important features, in order of strongest to weakest. (HF1) Jaro-winkler overlap; (HF2) Bell curve post time overlap; (HF3) Vocabulary size overlap; (HF4) Reply count overlap; and (HF5) Degree overlap.

4.5 Results from Hyper-Parameter Tunings

Because we operated on the the same forums for both the training and testing datasets, even though the users selected (and thus the user pairs) were separated, we took precautions for overfitting. Hyperparemeter optimization, or model selection, is one way to reduce overfitting. We both randomly sampled parameter combinations and observed all combinations of parameters, and used 10-fold cross validation (instead of randomly splitting our data) to choose the best model–the model that gave the lowest generalization error–for our data. We found the parameter combinations that yielded the lowest generalization error. We describe the results achieved from conducting hyperparamter tuning to find the best-fit parameters.

4.6 Grid Search Cross Validation

Grid Search is an exhaustive way to conduct hyperparameter optimization. It searches through all potential combinations of a set of specified possible parameters for the learning algorithms. The grid search algorithm's performance was measured by cross validation on the training set. The Grid Search top results, or the models with the highest ranking, for the hard training set can be seen in Table 2. Grid Search CV took 784.49 seconds, or roughly 13 minutes, and reviewed 108 candidate parameter sets.

4.7 GridSearchCV and RandomSearchCV scores

Classifier	params	val score	std
Random Forest	'bootstrap': True, 'min_samples _leaf': 10, 'min_samples _split': 10, 'criterion': 'gini', 'max_features': 3, 'max_depth': 3	0.819	0.006
Decision Trees	-	-	-

Table 2: Results Grid Search Hyperparameter Tuning using cross validation.

Random Search Cross Validation - Because GridSearch combines *all* possible parameter sets, it takes a significant amount of time and computing resources. Random Search is an alternative to GridSearch – Random Search samples a specified number of parameter combination candidates. It rivals Grid Search for two primary reasons: (1) A resource budget can be chosen and allocated independent

of the number of possible parameter combinations; and (2) Adding ineffective parameters does not affect the performance of the search. Random Search Cross Validation, therefore, takes much less time to perform with minimal loss of understanding parameter textures. Random Search took 175.39 seconds, roughly 3 minutes (a fraction of the time of GridSearchCV) and used 20 candidate parameter combinations. Random Searches top results can be seen in Table 3.

Classifier	params	val score	std
Random Forest	'bootstrap': True, 'min_samples _leaf': 7, 'min_samples _split': 10, 'criterion': 'gini', 'max_features': 8, 'max_depth': 3	0.817	0.018
Decision Trees	-	-	-

Table 3: Results Random Search Hyperparameter Tuning using cross validation.

focused on tuning parameters for random forests to find the appropriately balance between bias and variance. We chose the number of trees and the parameters to use by hyperparameter tuning with 10-fold cross validation. **Evaluation** - Results from the random forest classification can be seen in Table 4. The F1-score indicates the classifier's accuracy. True and false positives and true and false negatives can be seen to understand the classifier's precision and recall scores. Random forests did very well on the easy set, and maintained generalizably strong scores for the medium and hard sets.

-	Easy	Medium	Hard
F1-score	0.963	0.757	0.806
False Positives	3	10	30
True Positives	391	46	489
False Negatives	29	19	233
True Negatives	720	714	694
Total	1,143	789	1,446

Table 4: Results from random forest classifier after including hyperparameters.

5 CLASSIFICATION MODELS

We have continuous inputs (typically floats) and binary outputs (match or not match). We ran a decision tree and a random forest classifier. The classifiers and results will be explained in the following sections.

5.1 Decision Tree Classifiers

Trees are able to capture complex interaction structures, handle feature sets with high dimensionality, and represent the mathematical intricacies of the algorithm in a graphical, comprehensive way. We provided 14 pairwise features and a measurement of success for 722 observations, or unique hard pairs. We kept the default scikit-learn arguments set for Decision Tree Classifiers. So, the algorithm decided on the splitting variables, on the splitting points, and on the shape of the overall tree. A graphical representation of the tree for one cross-validation fold can found at [21]. **Evaluation** - The decision tree classifier had a 0.715 averaged f1-score across all 10 folds of the hard dataset. Trees are inherently noisy – they are known to have low bias and high variance. It appeared that a small change in the input data resulted in a vastly different split in the tree, as f1-scores ranged from 0.625 to 0.787 across the 10 training-testing splits. We used random forest classification, described in more detail below, to gain a more generalized result, and reduce the variance.

5.2 Random Forest Classifiers

To decrease the variance from Decision Trees, we observed how Random Forests behaved. Random Forests decrease the variance from Decision Trees through a process similar to bagging. The generalization error of random forests is affected as follows. The bias of random forests is similar to the bias of one single randomized tree, and randomization tends to increase bias. However, randomization of the ensemble model also tends to reduce variance. We largely

5.3 Challenges

Graph generation, graphical choices, choices involving whether to keep or not keep users who did not post, and the resources involved in calculating all of the distance metrics between all possible user pair combinations were difficult challenges that we faced. In terms of graph generation, given how contingent our graphs were on post-data, and how many users lacked post data, our graphs inherently eliminated all non-posting users. Moreover, we had to decide on choices of depth for each graph: would degreeness be egress or ingress edges? We chose both. Would we maintain metadata on how many hops each of the users were from one another in the same graph (on the same forum)? We chose: no. Would we maintain more information in the edge attributes? Our choice: we could have, and those could have lead to additional features, but for now, we tried to maintain simplicity. The resources involved in calculating the distance metrics were also taxing. For all user pair combinations across ForumA and ForumB, for only the users with post data, across 8 multithreaded processes on a MacBook Pro 10.10.5, calculations and classifications took 3.5 days. For all user pair combinations across Forum14 and Forum17, for only the users with post data, across a single-threaded process on a MacBook Pro 10.10.5, calculations and classifications took 10 days. We hope to accelerate this process in the future. Other papers [7] have explored creating subgraphs around target nodes to analyze the target nodes' functionality or behavior. We could similarly, for example, focus solely on users that are within a known match, and focus on pair combinations of that subset of users. Alternatively, we could have parallelized these calculations on a remote, high-CPU cluster. We could have also focused on summary metrics for each of the graphs, though graph-based (rather than user-based) metrics would help determine if two forums are similar, rather than if two users are identical.

6 FUTURE WORK

There are a number of areas of future work on our project. First, we plan on including separations of and clarifications of misppelled or conjoined words. Further we intend to better research approaches to weighing and separating non-dictionary words used by forum posters. We also will examine characteristics of white space and non-alpha-numeric characters such as punctuation while dealing with user post characterization. We will also take into account username mentions in posts, and finally we are looking into social circle characterizations and additional graph features to leverage to create the social circles evaluated by our approach.

7 CONCLUSION

We provided definitions of graph terms, definitions of stylistic analyses, and definitions of distance terms to analyze user-matches across different website forums. The majority of our initial experiments were conducted with a set of pairs, split 25% true matches, 75% false matches. This user-pair matrix allowed us to efficiently generate all features and distance metrics to analyze classification accuracies. We developed a Decision Tree, and a Random Forest Classifier (which performed best with an F-Score of 0.963), according to generalization across three sets of user matching problems: easy, medium and hard sets of username matches. It is easy to see how these techniques can be applied across other web forums and markets to find identical or highly similar users.

ACKNOWLEDGEMENTS

This effort was supported in part by JPL, managed by the California Institute of Technology on behalf of NASA, and additionally in part by the DARPA Memex/XDATA/D3M programs and NSF award numbers ICER-1639753, PLR-1348450 and PLR-144562 funded a portion of the work.

REFERENCES

[1] Ahmed Abbasi and Hsinchun Chen. 2006. Visualizing authorship for identification. *Intelligence and Security Informatics* (2006), 60–71.

[2] Juan J Alvarez, Florina Almenárez Mendoza, and Miguel Labrador. 2017. An accurate way to cross reference users across Social Networks. In *SoutheastCon, 2017*. IEEE, 1–6.

[3] Vincent D Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review* 46, 4 (2004), 647–666.

[4] W Ravikumar Cohen and P Fienberg. [n. d.]. A Comparison of String Distance Metrics for Name-Matching Tasks. In *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*. 73–78.

[5] Hao Fu, Aston Zhang, and Xing Xie. 2015. Effective social graph deanonymization based on graph structure and descriptive information. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 4 (2015), 49.

[6] Thamme Gowda, Kyle Hundman, and Chris A Mattmann. 2017. An Approach for Automatic and Large Scale Image Forensics. In *Proceedings of the 2nd International Workshop on Multimedia Forensics and Security*. ACM, 16–20.

[7] M. Hinne. 2011. *Location Approximation of Centrality Measures*. Master's thesis. Rabdoub University. http://www.ru.nl/publish/pages/769526/local_approximation_of_centrality_measures_-_max_hinne.pdf

[8] Laikhram Jamjuntra, Pantakan Chartsuwan, Peerapong Wonglimsamut, Kriengkrai Porkaew, and Umaporn Supasitthimethee. 2017. Social network user identification. In *Knowledge and Smart Technology (KST), 2017 9th International Conference on*. IEEE, 132–137.

[9] Fredrik Johansson, Lisa Kaati, and Amendra Shrestha. 2013. Detecting multiple aliases in social media. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*. ACM, 1004–1011.

[10] Fredrik Johansson, Lisa Kaati, and Amendra Shrestha. 2014. Time profiles for identifying users in online environments. In *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*. IEEE, 83–90.

[11] Bell Labs. 1971. Unix Programmer's Manual (PDF) First edition. (Nov 1971). https://www.bell-labs.com/usr/dmr/www/1stEdman.html

[12] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. 2013. What's in a name: an unsupervised approach to link users across communities. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 495–504.

[13] Qiang Ma, Han Hee Song, S Muthukrishnan, and Antonio Nucci. 2016. Joining user profiles across online social networks: From the perspective of an adversary. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 178–185.

[14] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. 2012. Studying user footprints in different online social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*. IEEE, 1065–1070.

[15] Chris Mattmann. [n. d.]. Tika Similarity - GitHub. ([n. d.]). http://github.com/chrismattmann/tika-similarity

[16] C. Mattmann, G. Yang, H. Manjunatha, Thamme N. Gowda, AJ Zhou, J. Luo, and LJ McGibbney. [n. d.]. Multimedia metadata-based forensics in human trafficking web data. *WSDM - SEXI Workshop* ([n. d.]), 10–13.

[17] Asitang Mishra. [n. d.]. Persona Linking - GitHub. ([n. d.]). https://github.com/asitang/PersonaLinking

[18] Erwan Moreau, François Yvon, and Olivier Cappé. 2008. Robust similarity measures for named entities matching. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 593–600.

[19] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[20] Martijn Spitters, Femke Klaver, Gijs Koot, and Mark van Staalduinen. 2015. Authorship Analysis on Dark Marketplace Forums. In *Intelligence and Security Informatics Conference (EISIC), 2015 European*. IEEE, 1–8.

[21] S. Stathatos. [n. d.]. Expanded Person Profile Tree. ([n. d.]). https://drive.google.com/file/d/0B_txn5f6YNusaUd3ZU5FeWxWcGM/view

[22] Jan Vosecky, Dan Hong, and Vincent Y Shen. 2009. User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT'09. First International Conference on*. IEEE, 360–365.

[23] Kuansan Wang, Christopher Thrasher, and Bo-June Paul Hsu. 2011. Web scale nlp: a case study on url word breaking. In *Proceedings of the 20th international conference on World wide web*. ACM, 357–366.

[24] Kuansan Wang, Christopher Thrasher, Evelyne Viegas, Xiaolong Li, and Bojune Paul Hsu. 2010. An overview of Microsoft Web N-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*. Association for Computational Linguistics, 45–48.

[25] William E Winkler. 1999. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer.

[26] Shayi Zhang. [n. d.]. LingCues–A Linguistic Cues Software Tool for Research in Text-based Automatic Deception Detection. ([n. d.]).

[27] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the Association for Information Science and Technology* 57, 3 (2006), 378–393.