Geographic Localisation of an Anonymous Social Network Message Data Set

Alexander Böhm, Benjamin Taubmann, Hans P. Reiser
University of Passau, Germany
Email: boehm@computer.org, {bt,hr}@sec.uni-passau.de

***Abstract*—Nowadays, privacy and anonymity are becoming more and more important for users of social networks. Thus, the degree of anonymity is of particular interest for them. In this work, we present one approach to obtain the geographic location of users in a popular anonymous social network. By using requests from different locations we can regain the location with an accuracy of a 10 meters with 20 requests.**

Keywords: Privacy, Deanonymization, Smartphone, Text mining, Machine learning

I. Introduction

Jodel[1] is an anonymous instant messenger for Android and Apple smartphones. It provides a live feed of messages located within a certain radius around the current position of the user. For those messages only a vague information of the location is displayed. Every user is able to contribute so called "jodels", anonymous messages, which can be seen by other users in real time. Jodel also provides a feedback instrument for users to up- and down vote those jodels (so called karma).

One reason for the popularity of Jodel can be explained with the concept of anonymous messaging. Lawler et al. showed that about 74.5% of the users within a social network mark their profile as private in a way such only friends and invited users can obtain the personal data [4]. Thus, it needs to be discussed whether applications such as Jodel can provide the promised anonymity. This can be done either by analyzing the infrastructure and find ways to exploit security flaws or by identifying data sources that allow gaining knowledge about the author of a message. In this paper we will focus on the second approach and discuss two datasources that can be used to achieve better knowledge of the sender which can be used to identify the author.

In contrast to prior work in that field, the public data which can be acquired does not provide any ground truth, i.e., no relation from a message to a user is given.

The research question of this paper is how to assign a message to a certain user or at least group of users when no ground truth is given. Additionally, messages are short (about 50 to 100 characters) which makes it hard to distinguish them from messages of other users.

[1]https://jodel-app.com, accessed 28 March 2016

In this paper, we present an approach to retrieve the GPS coordinates from the place where a message was submitted. Therefore, we exploit the fact that messages are only visible when the reader of the Jodel messages is in the same area where also the message was posted. To get the location, we send requests from different locations and check whether we can see the specific message or not. By using the approach we can get the original location of a message with an accuracy of about ten meters with about ten requests.

Our contributions are the following:

- The presentation of a practice approach to regain the original geolocation of anonymized locations
- An evaluation on how feasible this approach is for large amount of data
- A discussion about possible mitigation techniques based on provider side and techniques how users can increase their privacy against the presented attacks

The remainder of this paper is organized as follows. At first, in Section II, there is a review of some related work within this field of interest. After that, in Section III, we present our approach how to retrieve the geographic location of those messaged and what kind of machine learning is used to get an insight on the data set. A short sketch about the implementation will be given in Section IV. In Section V we evaluate the resulting data set, present our findings on it, rate the approach. Afterwards in Section VI we show ways to mitigate this kind of attack. Finally, we conclude the work in Section VII.

II. Related Work

This section gives an overview about related work on the topic of our paper. It is divided into three parts: At first it discusses Android application analysis, then it shows related work on generic deanonymization possibilities on social networks, and at last it focuses on the analysis and deanonymization of textual data sets. Deanonymization of Jodel is mostly about matching and mapping certain messages to unknown users. For further analysis, interactions in between the users can be profiled with the same approaches as for anonymized social networks.

A. Deanonymization of social networks

In many social networks, users can participate in an anonymous way, or data is later used after some

pseudonymization step. Several researchers have addressed the possibility of identifying individuals in data sets in which the identity is not explicitly present.

An attempt to deanonymize social networks based on the network topology is performed by Narayanan and Shmatikov [7]. The authors have shown that it is possible to reidentify the users between two social networks with a 30.8% success rate (12% false positives and 57% unidentified users) by an analysis of the social graph and its metadata. Since Jodel does not provide such a graph the data must be preprocessed (mapping messages to an anonymous sender) before analyzing with their methodology. If two unknown users tend to interact with each other, there might be an existing link in another social network. This behavior could help to deanonymize the users as described by Backstrom et al. [1]. Having the possibility to locate the anonymous Jodels and their probable users, Gambs et al. showed a way to re identify the users by mobility Markov chains [3]. Narayanan and Shmatikov [6] showed in their work that removing identifying information is not sufficient for anonymity.

Ding et al. [2] presented an overview about deanonymiztion attacks such as mapping or guessing based approaches. The approach Backstrom et al. [1] used to perform deanonymiztion of the users is a mapping based one. In such an approach the labeled background data is mapped against the released data of every user to reveal sensitive labels of users. A guessing based approach maps the released data matched against the labeled background data. Such an approach is used by Narayanan and Shmatikov in [6].

B. Text-based deanonymization

Rong Zheng et al. have developed a framework for authorship identification [11] by analyzing lexical, syntactical, structural and content-based features of user messages. Their approach identified the authors based on combination of those four features with an accuracy from 70% to 95% percent.

Okuno et al. developed a method for "Content-Based De-Anonymization of Tweets" [8], which allows identifying authors by matching the content of messages from Twitter with profile information from an author's résumé. Basically, the authors train a classifier to identify the user, because one user often tends to use the same words in different documents.

III. Approaches

For using techniques of machine learning, such as clustering and classification, it is necessary to have a valid ground-truth which can be used to validate those approaches. At first, this section presents an approach how to obtain the geographic origin of messages with a high accuracy. Afterwards, it shows an alternative approach by using methods of machine learning.

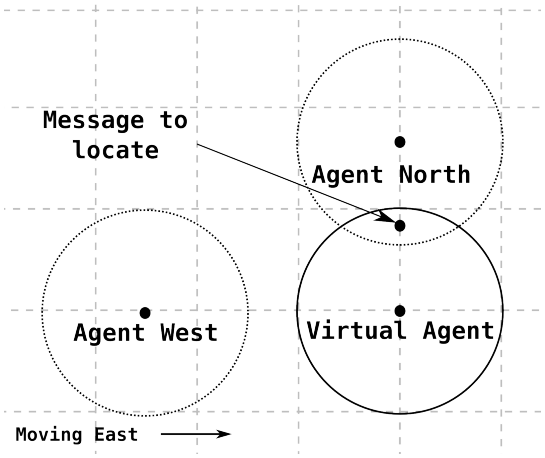

Figure 1. The process of locating a Jodel message, by moving two agents with known coordinates towards it until both retrieve it the first time.

A. Obtaining the Geolocation

This part describes how to obtain an accurate information about the coordinates from where a Jodel message was posted. Each message contains only the city and two decimal places of long- and latitude is sent from the server, which makes it impossible to obtain a precise geographic location of the user who submitted the message.

Each request for new Jodels contains all messages within a ten kilometer radius around the current location (local agent). This fact can be exploited to get the original longitude and latitude, by placing a virtual user (traveling agent) far north and west of the users position. Those two agents are slowly moved towards the local agent, until the message which has to be located is seen by both traveling agents the first time (see Figure 1). When the request of both traveling agents contain the message it can be located by calculating the coordinates of the intersection.

Instead of moving both traveling agents in small iterations to the virtual one, it is possible to perform a binary search by moving out of a ten kilometer radius, and then move towards the virtual agent by bisecting the distance between both agents: The traveling agent moves closer to the local agent until the Jodel is seen the first time. At this point it is necessary that the agent moves away from the virtual one until the Jodel is gone again. With each bisection the size of the steps the traveling agent moves from or to the location of the virtual agent is becoming smaller, making the location more precise. This bisection can be stopped either if the Jodel is located on the edge of the circle or a reconfigured inaccuracy is left. This approach performs like a binary search on a list, and thus reduces the number of requests that is required to locate a Jodel by the factor of $\log_2$.

B. Clustering / Textmining

A "Jodel" is plain textual messages without any categorization but sometimes users add a hash-tag (like Twitter)

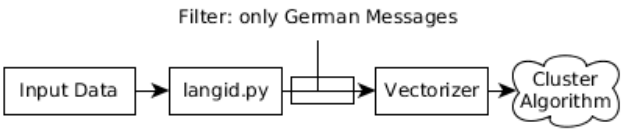

Figure 2. Data Processing Pipeline for Text Clustering.

to categorize their message or highlight a certain aspect. Clustering Jodels by their content provides insight about topics and keywords used in the messages. This work covers a short analysis of the messages by their content, their tags and relating the clusters to current events.

Natural language must be preprocessed before clustering. The required steps are as follows. First, each message must be vectorized and then feed into the clustering algorithm to obtain the clusters. Therefore, all non German messages will be sorted out. Then the features of a message are extracted. This processing pipeline is the same for clustering and machine learning (see Figure 2).

If each word should be weighted equal, the messages will vectorized by a bag-of-word-approach. It is recommended to use this approach for the analysis of hash tags or very short messages because it does not consider the term- and document frequencies in weighting the features. Messages can be vectorized by term-frequency inverse document frequency (TF-IDF). This vectorizer weights the term frequency current document and relates it to the frequency in all documents.

Furthermore, it is useful to remove a list of stop words from the messages prior to vectorization, in this work a list[2] of 129 German stop words is used. Jodel messages are very short, so it is sometimes not sufficient to weight words by TF-IDF-metrics.

After vectorization of the data, it will be clustered by a k-means algorithm into five clusters. Then their top-10 words are analyzed.

IV. Implementation

In this section we will present the implementation for each of the approaches described in the previous section. For obtaining the geolocation we will use a modified implementation of a public available API reimplementation for Jodel. For machine learning and text clustering, scikit-learn is used, the core functionality of this package is described by Pedregosa et al. [9].

A. Obtaining the Geolocation

For the acquisition of the dataset we used a modified version of JodelPy[3] which reimplements all necessary parts of the API in Python. Using this as basis, we created a modified multi-threading and multi-instance aware version. We modified the library in such a way that it supports our binary search approach and normalizes the returned data to a fixed

This version uses a database backend for configuration and data aggregation. For each city we use a dedicated grabber and locator thread so we are able to handle multiple cities at the same times. Information about gathered messages will be stored within a database.

Algorithm 1 shows how we have implemented our approach for obtaining the coordinates of a traveling agent (e.g. one that moves towards the virtual one from West to East). This is repeated for the second traveling agent which moves towards the virtual one from the other direction (e.g. from North to South). After obtaining both locations of the traveling agents, we calculate the intersection of the message horizons of both agents. The so resulting coordinates is the location of the message.

```
Input: Unique Identifier of Message
Input: Required Window Size
Input: Position of Traveling Agent 10000 meters
       away from Virtual Agent
Distance = 10000 meters;
Window Size = 1500 meters;
Minimal Distance = Distance - 750;
Maximal Distance = Distance + 750;
Middle of Distances = 0;
while Window Size is Greather Than Required
 Window Size do
    Window Size = Maximal Distance - Minimal
     Distance;
    Middle of Distances = (Maximal Distance +
     Minimal Distance) * 0.5;
    Move Traveling Agent to the Middle of
     Distances;
    Get All Messages at this Location;
    if Unique Identifier of Message is found then
        Minimal Distance = Middle of Distances;
    else
        Maximal Distance = Middle of Distances;
Output: Position of Traveling Agent
```

Algorithm 1: Implementation of our Binary Search Like Approach for Geolocating a Message in One Dimension.

B. Clustering / Textmining

For clustering we extract the messages for a given geographic area. The resulting messaged is filtered by langid.py [5] to sort out non German ones. As filter criteria, we are using all messages with a confidence greater than 0.7. Next we will vectorize all messages with the scikit-learn `TfidfVectorizer` and perform clustering by `KMeans`.

[2]https://code.google.com/p/stop-words/source/browse/trunk/stop-words/stop-words/stop-words-german.txt?r=3

[3]https://github.com/jafrewa/JodelPy

V. Evaluation

This section covers the evaluation of our approaches and describes the properties of the data set. First, we describe the data set. Aftwards, we present the results of the statistical analysis on the data set to get an insight on the community. Then we rate the results of our geolocation approach. Afterwards, we evaluate the results of the text mining and the machine learning approach. Finally, we discuss approaches to mitigate this issue.

A. Dataset

The data in this set was captured from 13th of January 2016 to the 20th of January and located within the same timespan. It contains about 38,000 Jodels whereas at least 96% have been located. Table I shows a detailed statistic about the distribution over the cities. Although Jodels gives user the possibility to upload images, this feature is rarely used: About 500 posts containing an image have been submitted.

Location	Total	Located	Percent
RWTH Aachen	4.672	4.439	95.01 %
TH Deggendorf	1.073	1.071	99.81%
TU Dresden	5,799	5,648	97.39 %
TU München	11,693	11,348	97.38%
Uni Bayreuth	5,783	5.575	96.40%
Uni Mannheim	3,775	3,694	97.85%
Uni Passau	5,543	5,366	96.80%
Overall	38,338	37,141	96.88%

Table I
Statistic over Cities

Location	Minimum	Maximum	Mean	Std.Dev
TU München	0	248	61	51
TH Deggendorf	0	240	67	53
Uni Bayreuth	0	248	66	54
RWTH Aachen	0	240	53	47
TU Dresden	0	248	63	51
Uni Mannheim	0	240	63	51
Uni Passau	0	240	60	51

Table II
Distribution of Message length in Chars per City

To get an insight about the data, user dynamics and possible contents we will have a short look into this data set by numbers and statistical means such as the message distribution per hour or the average length of a message. This information can be used to optimize further work on this community.

Figure 3 shows the distribution of messages during the day for each city in absolute number. At this point it is possible get an insight about the structure of the city and its community. Munich as capital of Bavaria has a very active community, resulting in high numbers of messages during the day. Overall, the amount of messages reflects the daily activity of the users which is posted as messages on this application. It is possible to get insights about the culture of a city by looking at the distribution of messages during the day.

The relative distribution over the length of messages can be found in Figure 4. The longest message is 248 chars long, the mean over all messages is 62.3 chars with a standard deviation of 51.7. The message distribution for each city is shown in Table II. The users of each city behave in the same patterns, because they use the service on in the same way and there is no city with tends to longer messages.

which picture In this picture there are two peaks, one at 48 characters length and the other one at 200 characters. The short ones contains a collection of poems, which have been posted during the caption time, the other peak is about someone reflecting his or her live.

B. Geolocation

To obtain a high accurate precision, we use the approach that we described in Section III-A. Our implementation supports a configurable resolution which is set to 10 meters in both directions (north-south and west-east). The binary search approach reduces the amount of messages by the factor of log_2, because it does not have to move the traveling agent for every single step towards the real location. The maximum iteration count to determine one dimension of a message is 13 requests. The measured results for our binary search approach are in average 10 requests per dimension. A usual distance flow between the traveling agent would fit to the pattern: 10km / 5km / 2.5km / 1.2km / 0.6km / 300m / 150m / 75m / 38m/ 19m / 10m.

To proof the accuracy, some Jodels where submitted by the API and then compared to the retrieved result. Therefor, we used the JodelPy reimplementation with a constructed location and a unique message.

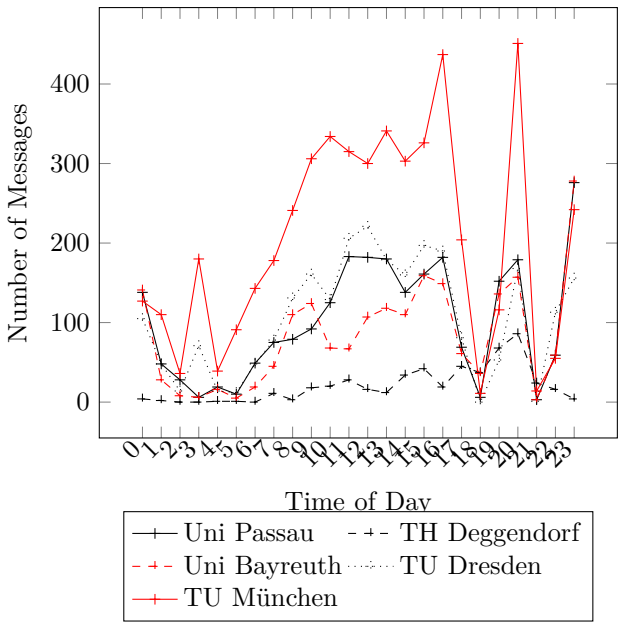

Figure 3. Absolute Distribution during the Day per City per Hour

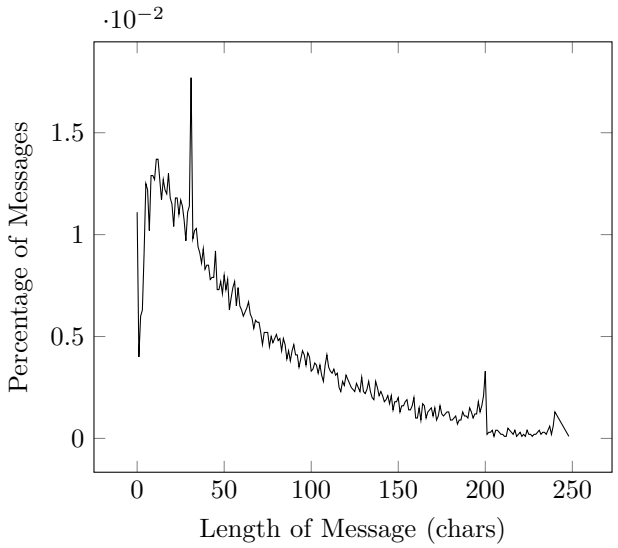

Figure 4. Relative Message Distribution over Length

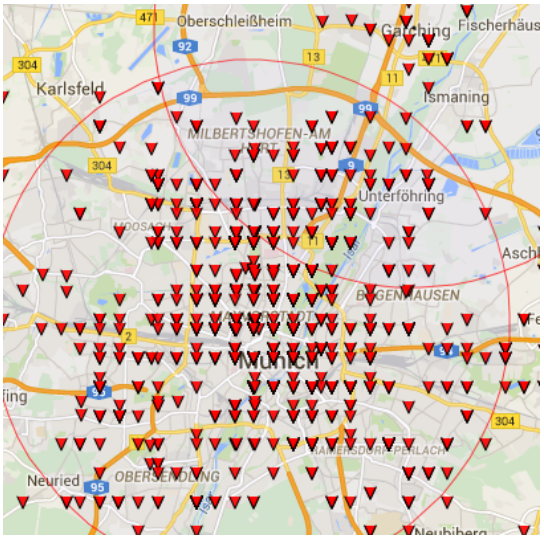

Figure 5. Distribution of Messages in Munich. Each Triangle represents a Message and the Red Area is Monitored. Source: Google Maps

Another validation was done by submitting Jodels at various places and note down their coordinates by a dedicated GPS-Device (Garmin GPS eTrex Vista HCx). The difference between coordinates on external device, the mobile phone and the located coordinates by our approach were within a five to ten meter inaccuracy.

Figure 5 shows the distribution of messages for Munich with the university as center location for the virtual agent. It is clearly shown that most of the messages are around the city center and central parts of the city. This distribution results from the usage of the application during social interactions and events.

This approach only handles a given message and ignores every other which is returned form the API. Speeding up the process could be done by storing all which can be retrieved at a certain location the process of locating a narrow message could be simplified.

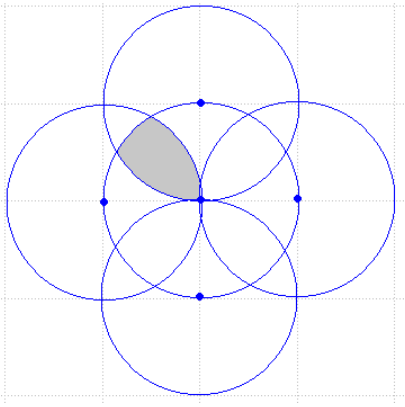

Figure 6. Monitoring a Certain Area by Placing Virtual Agents

By placing clients in such a way that the intersection of the seen message can be mapped to a certain area (see Figure 6). This would result in less accuracy but also reduces the amount of requests.

After locating a certain amount of Jodels those areas are given implicitly by selecting all messages witch has been seen by certain traveling agents and their known location.

Another way to improve the accuracy of this approach is to use a geographic dataset (e.g. OpenStreetMap) and map "floating" locations to a narrow hot spot such as a public building or a student housing. This would generate some inaccuracy over the real coordinates but would group the message around relevant social points.

C. Textmining

The differentiation between the messages of each city can be done by train a classifier: If such an algorithm can determine if a message belongs to a certain city and not to another, the content of a message or its structure is characteristic for this city.

If we can differentiate between two cities, we have shown that this approach can be also used for differentiate between two users. To get an insight into the possibility and have a larger data set, we decided to operate on city level first.

Clustering was done for three cities and their universities: Passau, Munich and Dresden. This selection was done because Passau represents a small city and university, whereas Dresden is a larger city and university, but all the main university buildings are located a narrow places. In contrast to that, Munich has one part of the universities within the city center and the other outside (Garching).

Clustering was done for five clusters, random initialization and a maximum number of 4096 iterations. Afterwards, the top ten keywords for each of those cluster have been extracted. For each of those keywords a random set

of messages have been reviewed to create a caption for this cluster.

It turns out that each city has a *social*-cluster within the top three clusters which contains messages about relationships or gossip. Messages in other clusters talked about *university life* or *television*. Overall the messages in those clusters represent non special day-to-day conversation topics.

VI. Mitigation

As shown in the previous section, performing geolocation is straight forward and can disclose information about a user which is intended to stay private. In this section, we are going to discuss techniques how to detect this kind of attack and how to mitigate it. This can either be done by an analysis of the client's behavior or by changing the data which is sent to a client.

A. Server Based Detection Techniques

Detection by the *number of client requests* is possible, but would lead to illegitimate exclusion for normal users. The approach described in this paper sends per Jodel per direction about seven to ten requests to the server, which is a common access pattern of a client. It might be possible to filter out this amount of requests but it would also lockout legitimate users which are behind a proxy or network-access-translation system. If there is a huge amount of clients, it might not be possible to detect those locating requests between the other ones.

Detection by *behavior* would lead to less false-positives because the characteristics of (binary-search like) requests can be easily spotted, because the virtual agent must be moved to the assumed location of the Jodel until its found. A possible way to prevent the detection is not to use North to South and West to East as directions for traveling agents but other directions. This behavior can be obfuscated in any other way, e.g., by emulating a walking person on a street and adding jitter.

As shown above, the detection of such a client is not easy possible because it would lead to false positives and or would lockout other users. Even if it is possible to block a certain IP-Address the attacker can chose a new one, which makes the previous block useless. If the attacker is using a proxy which is used by lots of legitimate users a lockout would cause a measurable dropout of users, because such proxies are used by internet service provider to map their customers to a single IP-addresses or to protect the network. Applying a network access translation is a common practice to encapsulate a mobile phone network from the internet and reducing the amount of used IP addresses and thus common practice by mobile ISPS. **was ist isps?**

At this point the question about caused load at the servers rises: as stated above it requires about 20 requests to determinate the location of a message. According to their own official news entry, Jodel has more than 100,000 installations [10]. If we assume that only 1% use it at the same time, there are about 1,000 current users, which is quite low for such a community. The refresh rate is about 5 seconds. Thus, there would be about 200 requests per second. The unoptimized approach would be 1% from the total number of requests for six new Jodels per Minute.

In this example it would be seen as constant amount which clearly stands out of the background noise of all the other requests, because this approach might have a backlog of high traffic times (e.g. evening) which have to be located during the night, before they vanish (as stated above only the last 60 messaged can be retried). Due to the fact that one of the IP addresses which have been used to create this data set have been black holed, it can be said that this constant amount of requests is traceable and noticeable.

Those possibilities might allow the mitigation of this attack but can be easily circumvented, e.g., by using various proxies, rate limiting or using different patterns to obtain the geographic location.

B. Adding Random Jitter to Coordinates

By adding a random jitter to the raw coordinates every time before they are used for calculating visible messages around the user might help to mitigate this attack. Unstable coordinates would lead to a higher inaccuracy by locating the message because it might disturb the binary search like approach. By configuring the jitter in such a way that it moves around within 50 meters would not affect the normal user because this person sees messages within a ten kilometer radius.

C. Grid Structure

Another way to mitigate this attack is to generate a grid around the surface of earth and delivery only the messages to users within the corresponding part of the grid. This would cause some side effects, e.g., if a user is between two cells, the GPS inaccuracy would cause that it will see messages either from one cell or from the other. Resolving this issue can be done by a fine structure of the grid and delivering all narrow cells of it.

A mitigation of this kind of attack is only possible if Jodel does not disclosure any metric about the distance between user and seen messages. At time of writing Jodel also provides a metric about the relative distance (indicated by a string within the application). This metric will be calculated by the server but could be easily reverse engineered. For a successful mitigation, this feature has to be adapted in a way so that it can not be exploited for this approach.

D. User based

In order to protect their privacy, users can set fake GPS coordinates for all applications on their smartphone. Thus, each Jodel message will be submitted from this fake position. However, a user should set a fake position for every new message. Otherwise, it is still easy to assign a message

to a user, when all messages have the same location. Or the location should be set to a public place where many Jodel message are submitted, e.g., the train station. Setting fake GPS coordinates is not only recommended for Jodel but for all applications that attempt to trace users by their location.

But this approach does not prevent against an authorship identification on machine learning techniques, because even if all users are on the same post, it is still possible to perform a text based user identification.

VII. Conclusion

In this work we presented one practical approach for obtaining the geolocation an anonymous social network message data set. We collected a data set of Jodel messages and created a statistical analysis of it. We showed how the anonymized geolocation of a message can be restored by testing if a message can be seen from different places while a message is only visible within the area where it was submitted. In our sample data set we achieved an accuracy of about ten meters per message for about 96% of the message.

By combining both approaches, we can assign a Jodel message to a person or at least a group of persons that live at the same area. By having this information we can build more complex models, train machine learning more intense and get better results.

References

[1] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[2] X. Ding, L. Zhang, Z. Wan, and M. Gu. A brief survey on de-anonymization attacks in online social networks. In *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pages 611–615, Sept 2010.

[3] S. Gambs, M.-O. Killijian, and M. Nunez del Prado Cortez. De-anonymization attack on geolocated data. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 789–797, July 2013.

[4] J. P. Lawler and J. C. Molluzzo. A survey of first-year college student perceptions of privacy in social networking. *J. Comput. Sci. Coll.*, 26(3):36–41, Jan. 2011.

[5] M. Lui and T. Baldwin. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics, 2012.

[6] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125, May 2008.

[7] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187, May 2009.

[8] T. Okuno, M. Ichino, T. Kuboyama, and H. Yoshiura. Content-based de-anonymisation of tweets. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*, pages 53–56, Oct 2011.

[9] 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*, 12:2825–2830, 2011.

[10] Gründerszene Magazin. Warum Studenten die App Jodel lieben und Promi-Investoren auch. http://www.gruenderszene.de/allgemein/jodel-app-erfolg, accessed 16-November-2015.

[11] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.*, 57(3):378–393, Feb. 2006.