DETECTING AND CHARACTERIZING ANOMALOUS FOLLOWERS ON SOCIAL MEDIA

by
BARIS TEMEL

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
December 2022

ABSTRACT

DETECTING AND CHARACTERIZING ANOMALOUS FOLLOWERS ON SOCIAL MEDIA

BARIŞ TEMEL

DATA SCIENCE M.S. THESIS, DEC 2022

Thesis Supervisor: Asst.Prof.Dr. Onur Varol

Keywords: Anomaly, Bot, Machine Learning, Twitter

This paper aims to detect anomalies in target social media accounts. Previous work on behalf of this topic includes bot detection applications with different types of methods. However, Anomaly detection is a more general framework that encapsulates social bots. Capturing anomalies starts with specifying possible anomaly types that can be seen in social media. In this study, we covered collective and point anomaly detection types targeting Twitter. Our algorithm includes features extracted from Twitter and synthetically created features which is the ratio of two or more other features. We create experiments to build an anomaly detection approach that can detect real-world examples.

Our study focused on both supervised and unsupervised machine learning models that capture the above anomaly types in an experimental environment. These models contain classifying algorithms. We used several scenarios to understand whether our model will be useful to use in a non-experimental environment, Twitter. We tested these scenarios under an anomaly ratio between 1%-10%.In conclusion, the experiments in this study have the purpose to demonstrate the outcomes of supervised & unsupervised anomaly detection techniques can capture these anomalous accounts.

ÖZET

SOSYAL MEDYADA ANORMAL TAKIPÇILERI TESPIT ETME VE KARAKTERIZE ETME

BARIŞ TEMEL

Veri Bilimi YÜKSEK LİSANS TEZİ, ARALIK 2022

Tez Danışmanı: Dr. Ögr. Üyesi Onur Varol

Anahtar Kelimeler: Anomali, Bot, Makine Öğrenmesi, Twitter

Bu makale, hedef sosyal medya hesaplarındaki anormallikleri tespit etmeyi amaçlamaktadır. Bu konu adına yapılan önceki çalışmalar, farklı türde yöntemlerle bot tespit uygulamalarını içermektedir. Bununla birlikte, Anomali tespiti, sosyal robotları kapsayan daha genel bir çerçevedir. Anormalliklerin yakalanması, sosyal medyada görülebilen olası anormallik türlerinin belirlenmesiyle başlar. Bu çalışmada Twitter'ı hedefleyen toplu ve nokta anomali tespit türlerini ele aldık. Algoritmamız, Twitter'dan çıkarılan özellikleri ve diğer iki veya daha fazla özelliğin oranı olan sentetik olarak oluşturulmuş özellikleri içerir. Gerçek dünya örneklerini algılayabilen bir anormallik algılama yaklaşımı oluşturmak için deneyler oluşturuyoruz.

Çalışmamız, deneysel bir ortamda yukarıdaki anormallik türlerini yakalayan hem denetimli hem de denetimsiz makine öğrenimi modellerine odaklandı. Bu modeller sınıflandırma algoritmaları içerir. Modelimizin deneysel olmayan bir ortam olan Twitter'da kullanılmasının yararlı olup olmayacağını anlamak için farklı senaryolar kullandık. Bu senaryoları 1%-10% arasında anomali oranı altında test ettik. Sonuç olarak, bu çalışmadaki deneylerin amacı denetimli & denetimsiz anomali tespit tekniklerinin bu anormal hesapları yakalayabildiğini göstermektir.

ACKNOWLEDGEMENTS

Words cannot express my gratitude to my professor and the chair of my committee for their invaluable patience and feedback. I also could not have undertaken this journey without my defense committee, who generously provided knowledge and expertise.

I am also grateful to my colleagues for late-night feedback sessions, and moral support. Thanks should also go to the librarians, research assistants, and study participants from the university, who impacted and inspired me.

Lastly, I would be remiss in not mentioning my family, especially my parents. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank my dog for all the entertainment and emotional support.

I dedicate my thesis work to my family and friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATONS

1. INTRODUCTION

Social networks foster connectivity among individuals. Those ties can emerge from real-world social networks, exchange of information, or mechanisms driven by homophily or mutual social ties (Weng, Ratkiewicz, Perra, Gonçalves, Castillo, Bonchi, Schifanella, Menczer & Flammini, 2013). Mechanisms controlling these networks may lead to more efficient and fair information dissemination (Wang, Varol & Eliassi-Rad, 2021) as well as promote the spread of misinformation (Shao, Ciampaglia, Varol, Flammini & Menczer, 2017). Similarly, the number of ties on a social network can signal popularity and prestige to some users, which eventually motivates malicious actors to manipulate those metrics for gaining visibility (Confessore, Dance, Harris & Hansen, 2018).

A popular practice to gain online visibility is purchasing fake followers. This approach artificially boosts follower count to create a popularity illusion and target the intrinsic biases of social media users. Social bots, compromised accounts, and 3rd-party applications are frequently used to manipulate network metrics. The New York Times journalists investigated follower factories used to promote influencers and celebrities. Similarly, researchers analyzed patterns of fake followers and how bot accounts can target popular journalists demonstrating different motivations to use fake followers among journalists (Varol & Uluturk, 2020).

Influencer markets are affected by this emerging fake follower problem. This market uses a social media marketing strategy. This involves an influencer creating content or promoting the brand to their community. There is a plague of fake followers, where influencers pump their follower numbers to deceive their clients. This study aims to find a payment policy without labeling fake followers of an influencer (Anand, Dutta & Mukherjee, 2020).

Fake followers are not limited to social bots. Other types of fake followers can be categorized as follow-for-follow, share-for-share, and like-for-like schemes. These schemes will only interact with the target if they complete an action that is in favor of their interest. These types of fake followers have a mutual relationship with the

target user. There are lots of online tools that a person with an interest to interact with these schemes can find by searching them for a targeted social media.

A social media platform called Twitter stated four truths about bot accounts on their platform. They are explaining various assumptions that an account looks to be suspicious, but is not in fact that account is fake or a bot account. They are stating these 4 truths that mislead people into wrongly classifying accounts as fake followers (Safety, 2008):

- Unusual account user names: Twitter states that bot accounts in their platform tend to have a combination of letters and numbers in their username. When it comes to detecting accounts looking by their usernames can misjudge the fact that Twitter also has name generating algorithm for the new user to quickly generate their usernames.

- Unusual user activities: There is no single way of using a social media platform. When people use a platform in different ways, they will see the opposite side as fake followers. In Twitter's case, they also referred to a change in how a user is using their platform varies by their culture, community, and nationality.

- Dehumanizing an opposite idea: Social media platforms creates a mixed bubble of ideas where you can see similar opinions of people interacting with each other more often. When two bubbles collide, accusations start to emerge that another side is not a real account. This behavior dehumanizes real accounts.

- Bot's interference in social media affecting real-world decisions: It is stated that whether fake followers or bots that are visible, spreading misleading information can change users' real-world decisions. It is hard to distinguish how we interact in social media and how we decide in the real world.

In this work, we present both supervised & unsupervised approaches to identify fake follower groups. We are cautious against getting biased detections in terms of the above 4 truths. The methodology developed in this study will lead to the automatic identification of fake follower groups, coordinated activities towards certain individuals, and uncover the anomaly event effects in social media.

1.1 Extended Summary of the Model

In this study, we have identified 4 main tasks: Follow Time Estimation, Synthetic Scenario Creation, Anomaly Detection, and Real World Use Cases

Our heuristic approach allows us to capture an estimated follow time for each follower of a target user. Additionally, we have created two synthetic types with various scenarios to mimic real-world anomalies, and our model aims to detect such anomalies with a higher detection rate.

We have placed a particular emphasis on parameter-free models, as these can identify anomalous behaviors without the need for specific parameter tuning, which is not optimal in real-world scenarios. Both supervised and unsupervised learning methods were utilized in our search for the optimal detection model.

Unsupervised methods were our initial approach, utilizing anomaly algorithms. These methods are particularly advantageous when studying real-world use cases. We also tested supervised learning approaches using the followers of each target user, along with the synthetically added anomalies. However, in real-world examples, this approach will be limited to identifying only the anomaly types that we have specifically called and trained for, and it may be vulnerable to noise in the training data.

1.2 Research Questions

In this study, we focused on several key questions before applying our approach to real-world scenarios. These questions are as follows:

- RQ1) Which features will be included in the model, do the chosen features represent real-world scenarios?
- RQ2) How to select the algorithms used in classifying?
- RQ3) How effective is the model created in the evaluation phase?

These research questions will be answered in the conclusion section.

1.3 Contributions

The primary contributions of our work to the problem of classifying social bots are the integration of follower anomaly estimation with modern machine learning models. Through this, we have made the following contributions to the above-mentioned problems.

- We estimate the time when a particular account follows our target account using public information provided by Twitter API.
- By extracting features from the follower's meta-data, profile information, and temporal patterns, we estimate the risk score for a follower being an inorganic follower.
- We demonstrate the effectiveness of our approach in a case study conducted in Political networks.
- The methodology developed in this study will be available online as a Python package and on a public repository.

2. RELATED WORK

There has been a growing interest in researching bots due to the rise of social bots (Ferrara, Varol, Davis, Menczer & Flammini, 2016), as their intelligence has begun to rival that of human beings. A competition called DARPA Twitter Bot Challenge has highlighted the importance of detecting bots. On the other side of the scale, a more recent competition sponsored/organized by Amazon is currently underway, with the goal of generating social bots that can engage in coherent conversation with humans for 20 minutes on a variety of topics (Staff, 2022).

Studying social media data comes with its own set of limitations. The extraction of features from this data is limited by the nature of social media applications. A study by (Zimmer & Proferes, 2014) found that roughly 5% of studies use existing Twitter Corpus collected from third parties instead of the Twitter API, and this number has likely only increased as access to the API has become more difficult over the years. The study identifies three key limitations of using Twitter data: representation bias, language challenges, and data bias (Ruiz Soler, 2017). They also note that Twitter has a low number of real follower usage compared to other social media applications, and outline the three methods for data retrieval using the Twitter API: Firehose, REST, and Stream. Both of these methods have two common issues: a cap on the number of tweets that can be extracted, and the fact that they only provide a real-time snapshot of the data, making it impossible to extract historical data.

The use of fake followers on social media has become a pervasive issue, with some individuals utilizing these fake accounts to manipulate and deceive others. This has led to the spread of false information and malicious attacks. The Botometer service, developed by researchers at Indiana University, offers a solution to this problem by providing a bot detection application that can identify the tactics used by specific bots to spread misinformation (Shao et al., 2017).

2.1 Anomaly Types

Over the years, anomalies have been classified into three main categories. A survey by (Chandola, Banerjee & Kumar, 2009) explored the nature of each category, highlighting the importance of distinguishing between the different behaviors of these categories. It is important for any model to accurately comprehend real-world scenarios where combinations of these anomaly types may exist in the data. Figure 2.1 is an example of the various anomaly types that may be present in a social media dataset. The x-axis represents the rank order of followers who have followed a target user, while the y-axis represents the creation times of each follower. In the dataset, we have added two anomalies, indicated by the blue and red points, which are homogeneously distributed among the real followers.

Little work has been performed in providing point anomaly detection. Point anomalies are the simplest type of anomalies. A point anomaly occurs when a single point is far outside the density distribution of the dataset. For example, in a lottery event, the person who wins the lottery can be considered a point anomaly if it is assumed that only one person wins out of all those who participated. Figure 2.1 shows an example of a point anomaly when the follower creation time is earlier than in 2011. These point anomalies were not added by us, as they exist naturally in the dataset without our manipulation. One study refers to point anomalies as "single anomalous windows" and detects them by maximizing the penalized saving by using a pruned dynamic programme (Fisch, Eckley & Fearnhead, 2022).

Collective anomalies have more than one point in the dataset having the same behaviors which are far outside the normal distribution. These anomalies are not shown anomaly behavior like point anomaly when there is only a single data point. It needs to be a collection of data points to create this anomaly type. One study used LSTM-RNN architecture to detect collective anomalies(Thi, Cao & Le-Khac, 2018). They tried to detect collective cyber security attacks at the collective level. Another study focuses on detecting anomalies using One-Class SVM for control functions of Hybrid Electrical Vehicles(HEV)(Ji & Lee, 2022). In Figure 2.1, we can see two different behaviors of collective anomalies. The first one is in the middle location of the data set colored red. They show group behavior that is different from their neighbors. These accounts are created very frequently with a variance of 6 months. The second contextual anomaly occurs at the end of the data set. These followers are recently created and followed by that target user within a period.

Contextual anomalies are the anomalies that are an anomaly within a specific con-

text. Point and Collective anomalies can easily be considered contextual anomalies if any mutual information is found between each data point type. Contextual anomalies are most generally studied on time series and spatial data. One study proposed a new method called NCAD to detect contextual anomalies on time series data(Carmona, Aubet, Flunkert & Gasthaus, 2021).

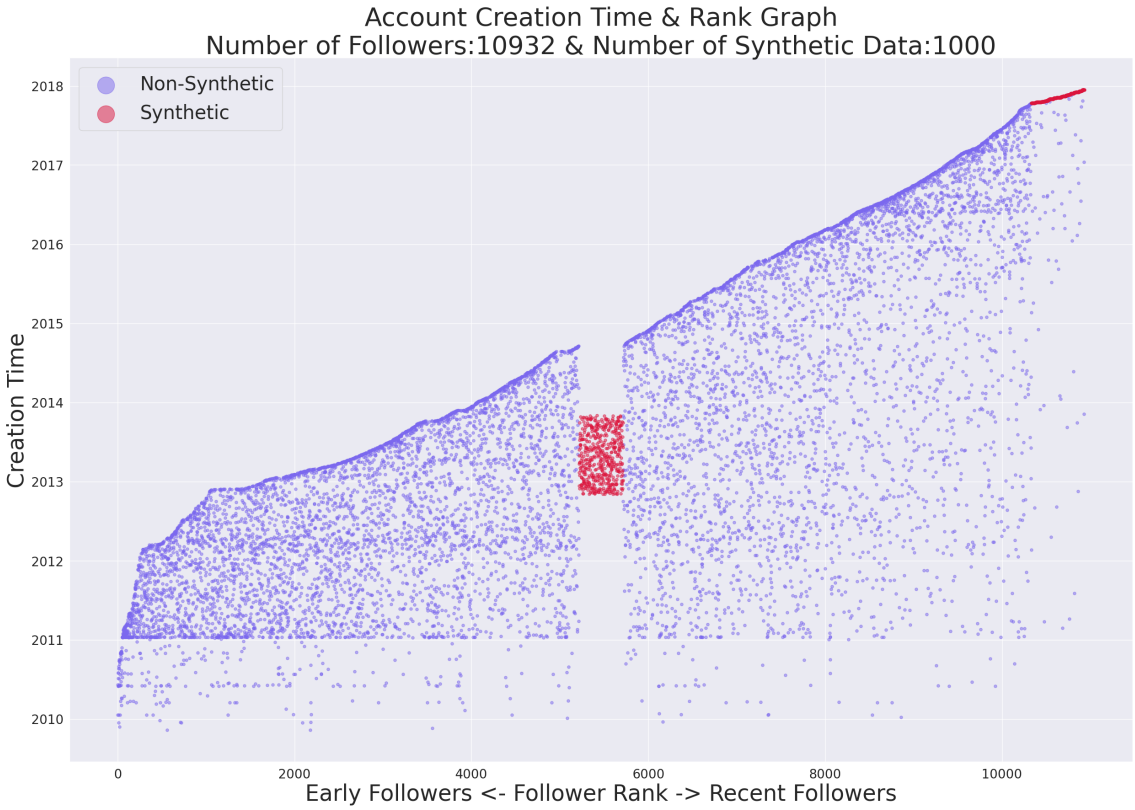

Figure 2.1 Anomaly Types Figure: Example representation of a target's followers with the added synthetic followers labeled as red.

2.2 Anomaly Detection

Anomaly detection is one of the most common uses of machine learning algorithms. If we compare manual detection where an auditor takes his time to inspect every anomaly occurrence and labels the data as an anomaly or not, it will take too much time. Machine learning models will have the advantage to combine statistics with model learning. Handling unstructured data is a straightforward task in machine learning models. Both unsupervised and supervised methods are used for anomaly detection.

General machine learning techniques about anomaly detection can be useful in several case studies as follows:

- Detecting effects of special events
- Deep understanding of bot behaviors
- Fraud Detection
- Intrusion Detection
- Image Processing
- Anomaly Detection in Text Data

2.2.1 Detecting Effects of Special Events

One study applies a clustering-based algorithm to detect unusual events in urban areas using Instagram's geolocated data(Domínguez, Redondo, Vilas & Khalifa, 2017). By utilizing DBSCAN to cluster urban areas, the researchers were able to detect high densities on special dates such as Christmas. (Garroppo & Niccolini, 2018) utilized cellular data to identify social events, using the SAGC algorithm to detect abnormal traffic peaks. This allowed the researchers to identify peaks on football match days, transportation strikes, Christmas, and the Politecnico of Milan on Sundays. A more recent paper focused on detecting the impact of COVID-19 on health-related tweets in Twitter text data (Kumar, Khan, Hasanat, Saudagar, AlTameem & AlKhathami, 2022). They used the BERT model's similarity score to detect a tweet as anomalous. Our anomaly detection pipeline can detect special occasions where a target user can have an unusual type of followers during a special event. These special events can occur at any time and can affect its anomaly ratio drastically. Such events can be special days for that target. For example, a politician can have an unusual increase or decrease in their followers right before the election times. A financial consultant can have a drastic change in their followers during economic events.

2.2.2 Deep Understanding of Bot Behaviours

Besides the descriptive approaches utilized through earlier years, work about detecting anomalies and social bots have shifted to machine learning models. Thinking of the concept of social bots, we can state that an account behaves like an automated agent that can mimic a real person's behavior (Varol, Davis, Menczer & Flammini, 2018). One study creates a detection model that classifies malicious accounts using generated account names and creation time (Lee & Kim, 2014). Another study tries to reverse engineer the social bots in Twitter (Bello, Heckel & Minku, 2018). They aim to find the behaviors of bots by observing real bot accounts. Bots have an effect on special events such as elections (Bruno, Lambiotte & Saracco, 2022), financial campaigns (Chen, Gao & Zhang, 2021), and COVID-19 discussions (Zhang, Qi, Chen & Liu, 2022).

2.2.3 Fraud Detection

Fraud continues to plague numerous industries, with the number of incidents on the rise each year (Abdallah, Maarof & Zainal, 2016). One study for Fraud detection uses SMOTE to solve highly imbalanced dataset problems (Varmedja, Karanovic, Sladojevic, Arsenovic & Anderla, 2019). They found that RandomForest algorithm gives the best result from their experiments. Another study suggests Quantum Computing(QC) for fraud detection models (Kyriienko & Magnusson, 2022). They compared both supervised and unsupervised methods of SVM's using Quantum Computing. With an increased number of features, the researchers discovered that Quantum Computed SVC and SVM were superior to classical algorithms.

2.2.4 Intrusion Detection

Another study for anomaly detection focuses on intrusion detection (Omar, Ngadi & Jebur, 2013). Intrusion detection is a monitoring system that tries to detect suspicious activities and alert the user. They state that when using state-of-art models, algorithms will have pros and cons. They come to the conclusion that KNN, One Class SVM, and SOM have better results than the other unsupervised techniques. Supervised learning methods tend to have a better detection rate. However, there

is an issue to create a training dataset that covers all intrusion detection areas. Another issue is there will be noise in the training dataset which will produce false alarms.

2.2.5 Anomaly Detection in Text Data

One study focuses on topically anomalous tweets and proposed a new method to assess the quality of a tweet. They calculate the quality by checking whether a claimed topic is indeed the actual topic stated in the URLs in the tweet (Anantharam, Thirunarayan & Sheth, 2012). Another article detects contextual anomalies in text data (Mahapatra, Srivastava & Srivastava, 2012). Another article suggests a new approach to detecting system faults from log text data (Shao, Zhang, Liu, Huyue, Tang, Yin & Li, 2022). Their algorithm is called Prog-BERT-LSTM. This study suggests a supervised method using an auxiliary dataset of outliers to train Anomaly detectors (Hendrycks, Mazeika & Dietterich, 2018). They suggest their method as Outlier Exposure. They tried anomaly detection both in text & image datasets.

2.2.6 Anomaly Detection in Image Processing

Work on anomaly detection techniques in image processing is very limited. (Chang & Chiang, 2002) studied hyperspectral imagery classification using two anomaly detection models called RXD and Uniform Target Detector. A survey combines all the image classification methods used in the literature, and groups them by the machine learning architecture used in the paper (da Costa, Papa, Passos, Colombo, Del Ser, Muhammad & de Albuquerque, 2020). Recent studies like (Hao, Li, Wang, Wang & Gao, 2022), (Patrikar & Parate, 2022), and (Ullah, Ullah, Hussain, Muhammad, Heidari, Del Ser, Baik & De Albuquerque, 2022) also focused on video anomaly detection models.

We will cover the first 2 aspects of case studies using Twitter social media. Comprehensive studies can be created using our tool to create a similar model that works in any other social media.

2.3 Follower Time Estimation

This paper achieved a notable milestone with its development of a heuristic approach that estimates the time at which a user begins following another user on Twitter. One previous study created such an estimation utilized for Twitter followers (Meeder, 2011). Borrowing from previously published methods, this paper demonstrated a method for Twitter's social network that takes a single static snapshot of network edges and user account creation times to accurately infer when these edges were formed. To test their approach, the researchers crawled the most recent 5,000 followers of 1,800 celebrities using the Twitter API, creating a sequential list of users and the time intervals when the followings occurred.

Figure 2.2 shows their Upper Bound to estimated follow-time errors for all their target users. They find that this error decreases from hours to minutes as the target user has more followers. This shows their method is accurate when there are high follow rates. The researchers also added historical accuracy for accounts with lower follow rates, and found that three events in particularly affected celebrity follow rates: the introduction of the suggested user list, updates to that list, and the addition of the "users you may be interested in" option to Twitter's interface. Another interesting aspect about following a celebrity is when a user follows a celebrity within a month of its creation time, it is most likely that the user immediately follows the celebrity while joining Twitter. The researchers also studied the relationship between increases in celebrity followers and real-world events, creating a relative popularity score that tracked the top 10 celebrities over the years. This analysis revealed correlations between various peaks and drops in relative popularity and real-world events. However, while such an approach has its advantages, it has not been combined yet with anomaly detection.

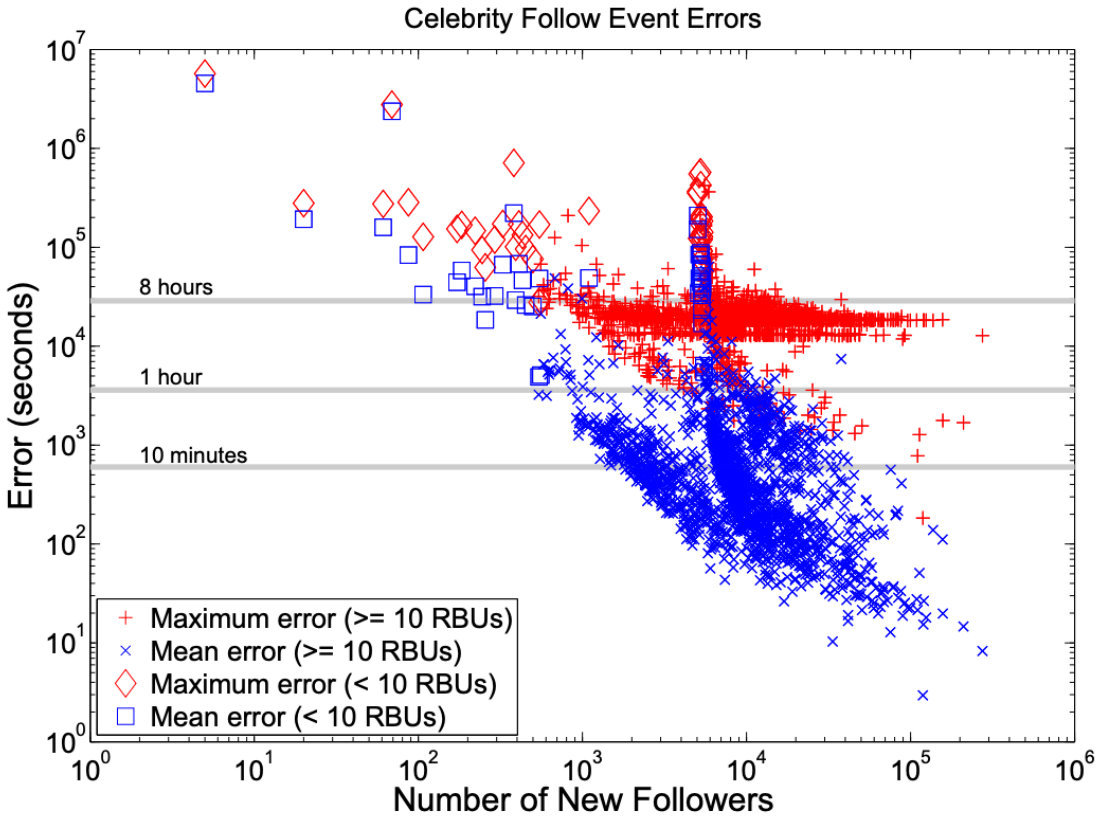

Figure 2.2 Celebrity Follow Errors Figure(Meeder, 2011)

3. DATASET

In this chapter, we discuss the datasets utilized in this study, focusing on Dribbble, and Twitter. We provide detailed statistics and visualizations to enhance the understanding and impact of the insights derived from the data.

3.1 Dribbble Dataset

A social media called Dribbble was used for evaluation. It is a self-promotion and social network platform for digital designers and creatives. It is established in 2009 and with 12 million users, it is the world's largest platform created for designers. The use cases of Dribbble can change for different people. Designers want to create and distribute their portfolios. Freelancers want fresh leads and opportunities and teams want to build pipeline for a collaborative work.

This dataset offers follow times as its label, allowing us to compute the mean error of the model for each target user. In this way, the model's mean error was created for each target user. The dataset contains 771.534 user information and 772.074 user-follower connections where the exact follow time is given. Our study collected users using random sampling to create an even distribution between targets that have low & high numbers of followers. We binned users that have followers in the range of 1,000-5,000, 5,000-1,000, and 10,000+. We collected users correspondingly 225, 225, and 50 from the whole dataset a total of 500 users. The distribution of the dataset is shown below the Figure 3.1. The third bin comprises a smaller number of users due to the limited availability of those with over 10,000 followers. In addition to follow times, the dataset also includes features such as shots count, projects count, biography information, username, type, number of likes received, number of followers, account creation time, comments received, and location information.

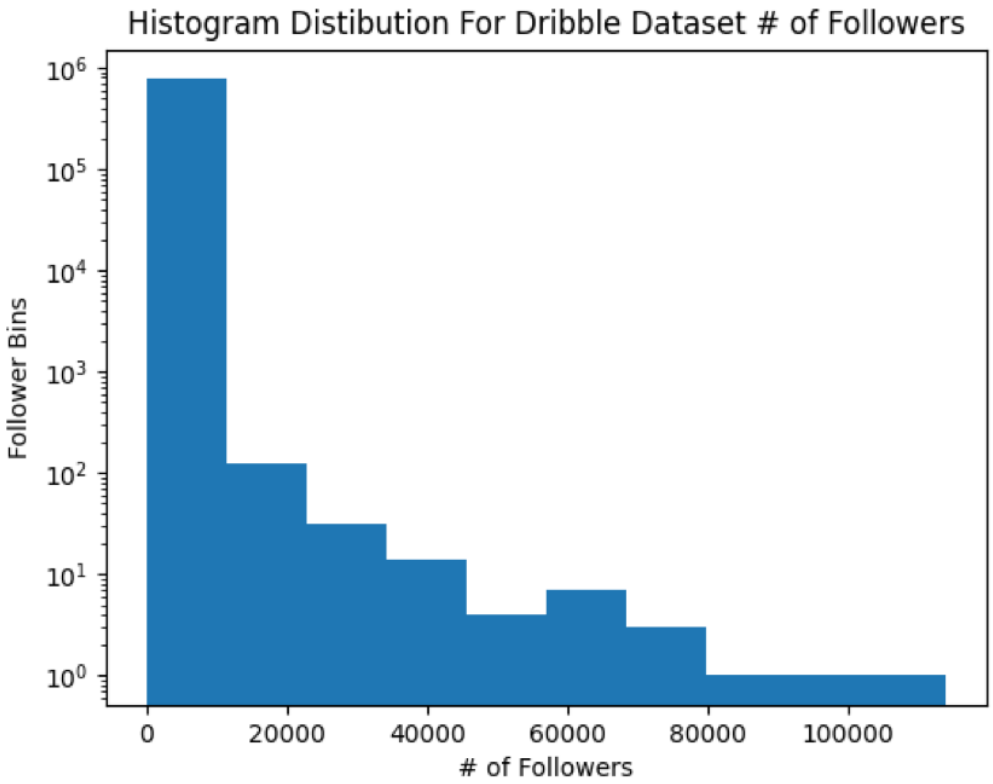

Figure 3.1 Dribbble Dataset Distribution

3.2 Twitter Dataset

Twitter is a social media site originally started as an SMS-based communication platform, founded in 2006. The initial idea was only for providing an internal service for an American company. Later, it evolved into a platform that spreads information very easily and fast. Its use is open to everyone, and the way it is used varies for everyone. There are approximately 237.8 Million active users that use Twitter daily.

We are using a backend Twitter wrapper API called Tweepy. This API can collect data by sending requests in the format of usernames or hashtag queries. We get various features of each username target. Each target JSON object is comprised of 3 fields:

- Creation Time

- User ID
- Rank

In addition to the following times that serve as the primary focus of our analysis, our dataset also includes features such as follower and like counts. Our system can be expanded to incorporate features from profile meta-data.

The data collection process is subject to strict rate limits imposed by the Twitter API. Unfortunately, Twitter does not provide follow times for any user, necessitating our evaluation of the Dribbble social media dataset.

4. METHODOLOGY

In this chapter, we discuss about how the study utilized the Dribbble dataset introduced in chapter 3, including a comprehensive overview of our experimental pipeline and visual results. We will also discuss the features employed and introduce the tested models.

4.1 Estimating Follow Time

The Twitter API provides followers in chronological order, allowing us to employ an empirical approach to calculate the lower and upper bounds of follower times. To determine the lower bound, we group users who followed the target prior to a given follower, taking the cumulative maximum dates of this group as the lower bound for the selected follower. Meanwhile, the upper bound is calculated by utilizing the creation times of the nearest neighbors. The final follow-time estimation is derived by taking the mean of these lower and upper bounds.

We can formulate this approach by the following steps:

- C_t = Current Follower's Creation Time
- P_t = Previous Follower's Cumulative Creation Time
- P_{t+1} = Next Follower's Cumulative Creation Time
- $LB = \begin{cases} C_t, & \text{if } C_t > P_t. \\ P_t, & \text{otherwise.} \end{cases}$
- $UB = P_{t+1}$
- Estimate = $(LB + UB)/2$

The process of generating an estimated follow time for each target user is carried out using the above formulation. It is important to know account creation times and rank to successfully create estimates. Figure 4.1 shows mean errors of estimated follow times in all 500 target examples of the Dribbble Dataset. The result is similar to Figure 2.2. As the number of followers increases, our estimated error decrease to smaller than 6 hours, and the model detection will be enhanced. The maximum error observed is 21 days and the minimum error is 3.11 hours with a number of followers of 1,064 and 97,973. The mean error is 5.45 days with a standard deviation of 4.46 days.

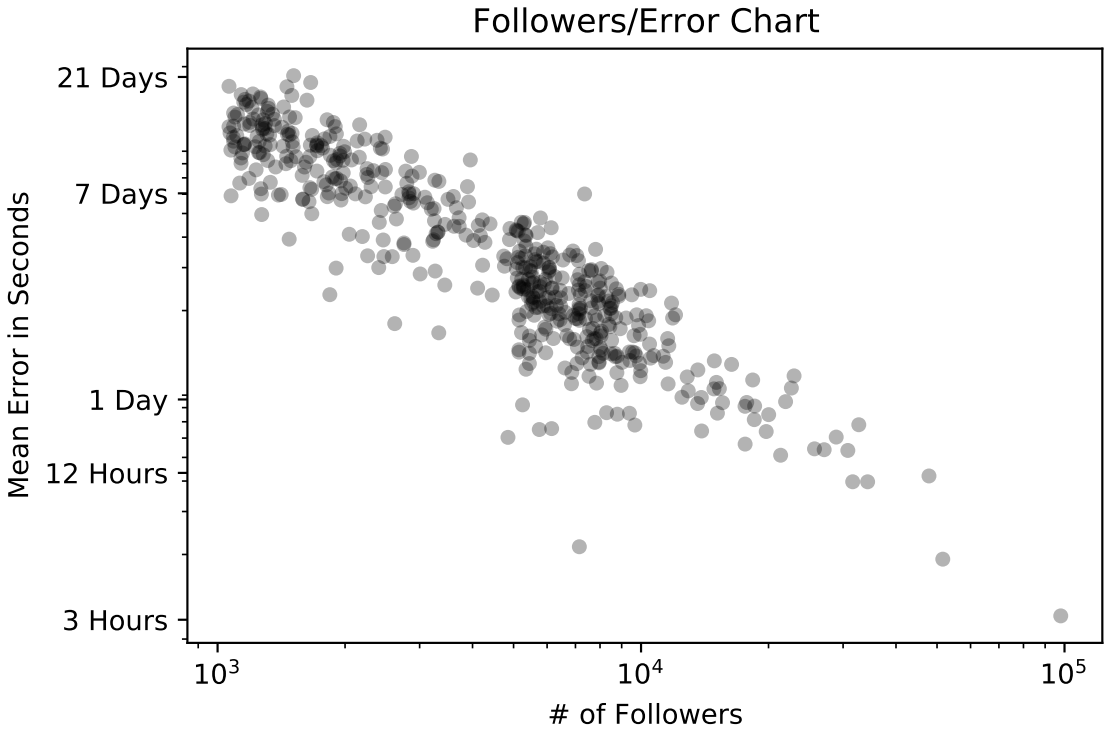

Figure 4.1 Dribbble Follow Errors Figure

Figure 4.2 presents where the median error occurs for relative ranks among all target users. Relative rank is calculated by using the rank of each follower user divided by the followed target's number of followers. From the calculation Relative rank is a number between [0,1]. In the figure, we can see earlier and last relative ranks tend to have higher mean errors.

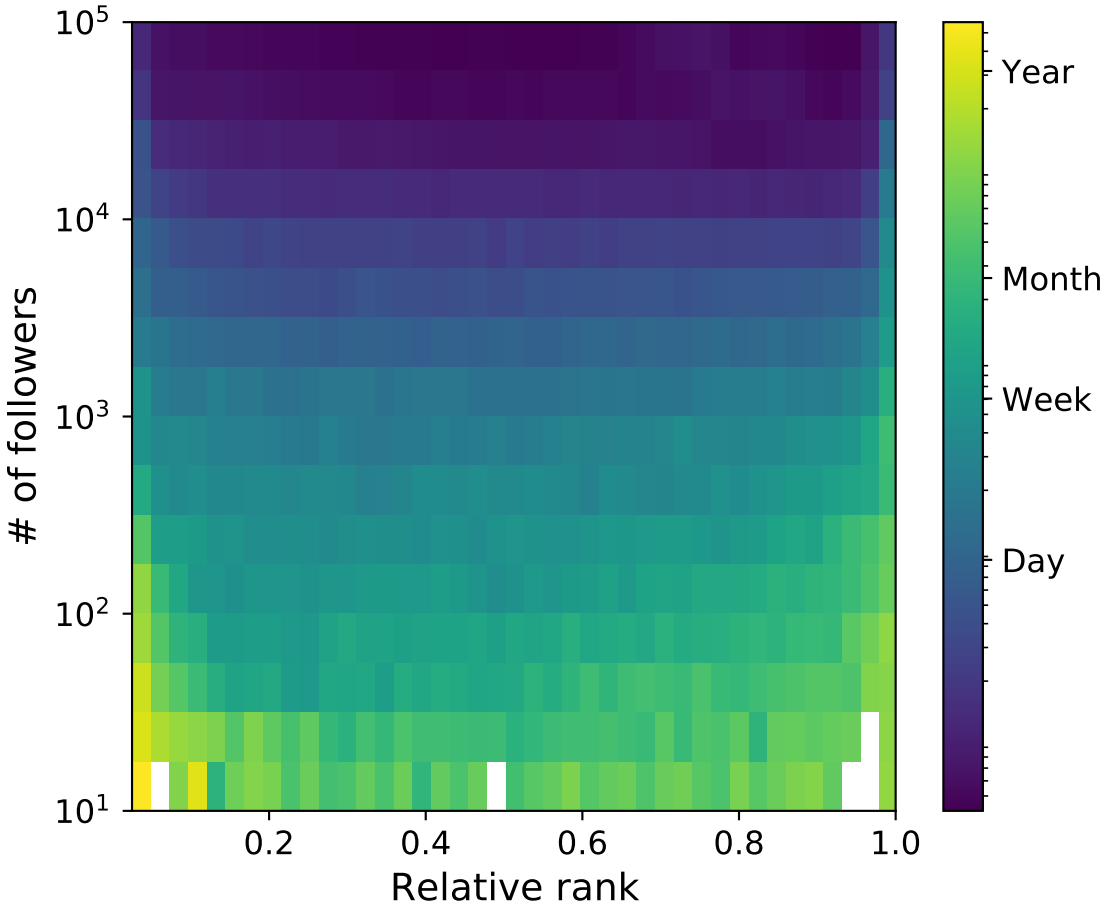

Figure 4.2 Dribbble Relative Rank-Median Error Figure

4.2 Follower Feature Extraction

In this study, there were 2 types of features, one of which directly came from Twitter API. Creation time & rank. The second type artificially created new features from the first type.

Twitter does not provide an exact time when an account follows another. However, we can collect followers of an account and this request returns a chronologically ordered list of users. Our model estimates a follow time for each follower using the account creation meta-data and the ranking in the follower list. Later we use the following time to create various features. Our model consists of 10 features. In table 4.1 you can find details about these features.

Table 4.1 Feature Definitions Table

Features		...Definitions...
Estimate(EST)	⇔	Estimate follower time for each user following the target.
Lower Bound(LB)	⇔	Lower bound of follower time for each user following the target.
Upper Bound(UB)	⇔	Upper bound of follower time for each user following the target.
UB-Creation	⇔	The difference between upper bound and creation time.
LB-Creation	⇔	The absolute difference between lower bound and creation time.
Relative Rank(RELRANK)	⇔	The current rank divided by the total number of followers for each user. This gives us a weight of relative ranks scaled by 0 to 1.
Frequency	⇔	We calculate the monthly frequency the followers are created.
Target – Neighbor	⇔	It is the difference between each follower's creation time and their 10 neighbor's mean creation times.
Neighbormax – min	⇔	It is the difference between the maximum and minimum creation time of each follower's neighbors.
Neighbor - mean	⇔	It is the mean between the previous 5 and next 5 creation times of each follower's neighbors.

We then check for feature correlations to decrease redundancy and improve both model runtime and evaluation performance. Below Figure 4.3, we show correlated features in a heatmap. Red squares show a higher correlation between pairs of features.

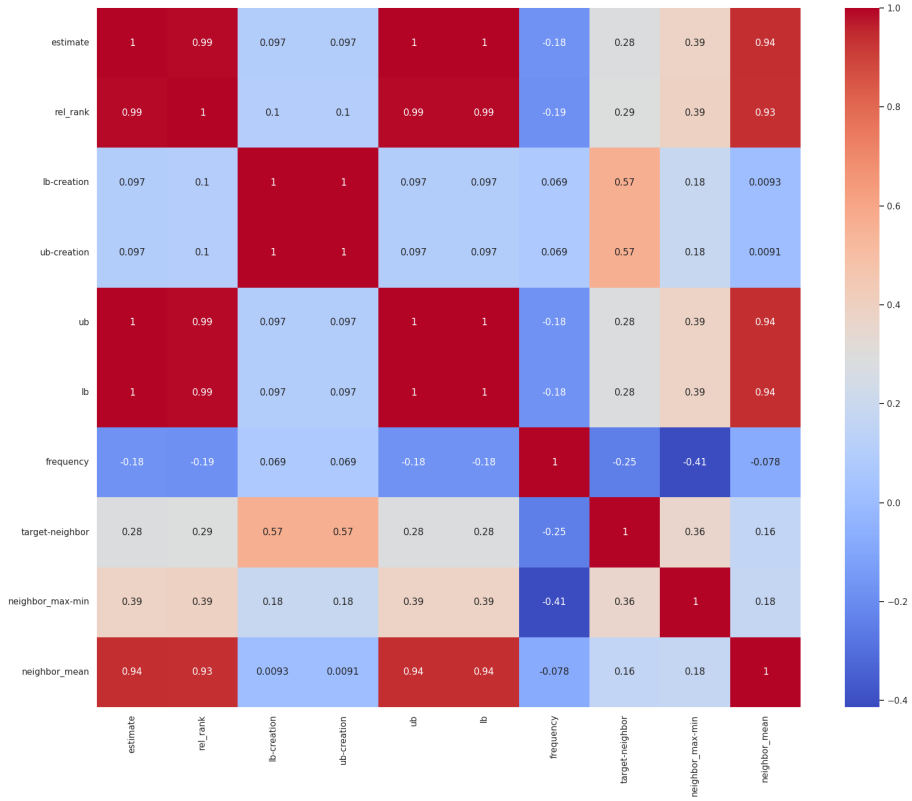

Figure 4.3 Feature Correlations Figure

We identify high correlations between EST and LB, UB. This is expected since EST is calculated using LB and UB. That is why we removed LB and UB from our feature matrix. We also see high correlations between lb-creation and ub-creation. We picked lb-creation and removed the ub-creation feature. There is also a high correlation between rel-rank and neighbor-mean but these features are useful in different user examples. To that end, we also checked the feature importance of these features shown in Figure 4.4. Feature importance show that for this example user which features contribute to the outcome of the model. Neighbor-mean and frequency changes for different examples. That is why we kept both of them in our model. At last, we have 7 features left in our model.

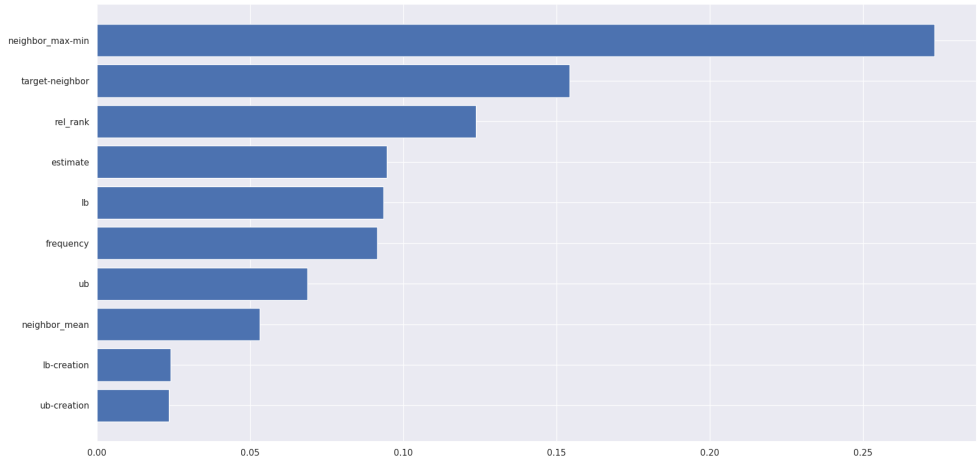

Figure 4.4 Feature Importance Figure

4.3 Models

After the feature extraction stage, we can test both supervised and unsupervised approaches using Dribbble dataset and synthetically generated instances.

4.3.1 Unsupervised Learning

In this learning pipeline, algorithms have all the target user's followers as the input. The algorithm has no knowledge of output labels both in training and the test phase. We used various anomaly detection models provided in pyOD package and HDBSCAN library. pyOD library has 30 anomaly detection algorithms that are useful to our goal. We have eliminated them to keep the best algorithm: the OCSVM algorithm. We used Feature Bagging Classifiers for each algorithm that improves the performance and accuracy metrics.

4.3.1.1 OCSVM

One-Class Support Vector Machines(OCSVM) are robust algorithms for classification and regression types of problems. Unlike state of art linear classifiers, SVM creates a vector between the extreme points of classes where their distance to the vector is almost equal and the maximum distance. This creates a best-fitted line of vector and classifies the dataset. One Class SVM has a similar approach but instead of creating a support vector between two or more classes, it creates a hypersphere. It states that if the training point is inside the hypersphere, it is an inlier. Otherwise, it considers that point to be an outlier. The continuous outlier score is measured using how far that distance is from the center of the hypersphere.

4.3.1.2 ECOD

Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions(ECOD) is a parameter-free algorithm. Their approach focuses on empirical CDF functions. It is a probabilistic type of outlier detection model (Li, Zhao, Hu, Botta, Ionescu & Chen, 2022). It is a fast and computationally efficient algorithm.

4.3.1.3 HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is an algorithm that performs varying epsilon values of DBSCAN and combines the results to find the clusters(McInnes & Healy, 2017). DBSCAN is a clustering algorithm that has two critical parameters called epsilon and minimum samples. HDBSCAN is a more robust way of finding clusters. It also supports a library called GLOSH outlier detection algorithm that detects the local outliers in the clusters (Campello, Moulavi, Zimek & Sander, 2015).

4.3.1.4 Isolation Forest

This model isolates data points by selecting features and sub-samples randomly. It is based on a decision tree algorithm. Anomalies have the shortest path in the trees compared to normal points.

4.3.2 Supervised Learning

In this learning pipeline, We split the targets into 80-20% ratio. We used cross-validation with 5 folds to accurately estimating out of sample accuracy and increase the efficiency of data usage as every observation will be used both for training and testing stages. We used SVC as our supervised model. We removed relative rank feature in supervised learning models in order to prevent overfitting.

4.3.2.1 RandomForestClassifier

RandomForestClassifier is an ensemble learning method for classification, consisting of various decision trees. It uses bagging and feature randomness to create each tree. Each individual tree predicts the class and the highest-voted class will be selected as a prediction.

4.3.2.2 SVC

C-Support Vector Classification(SVC) is a Supervised SVM algorithm. As explained in 4.3.1.1 one-class support vector machine creates a hypersphere for its decision function. The only difference is that in the training phase SVC also uses labels to learn about the anomaly detection problem.

5. RESULTS

In this chapter, we will discuss various case scenarios that we might encounter in the real world. We will prepare our model using Dribbble evaluation scores. Both Supervised and Unsupervised learning algorithms are studied. Then using these experiments we will discuss evaluation metrics. For the final part, we will discuss the real-world use cases of this study.

5.1 Synthetic Data Creation

Synthetic data is created for evaluating our model. This study conducted tests for the first, second, and combined types of synthetics and their different combinations of parameters. The purpose of synthetic data creation is to evaluate our unsupervised and supervised models. In real-world examples, we can not be certain that targeted social media has a certain number of anomaly accounts. Therefore we used this dataset and its homogeneity to conduct these studies. We create synthetics for all 500 target users we selected using random sampling from the Dribbble dataset. Locations of synthetics were kept the same for experimental purposes. We consider these synthetics will create an anomaly-like behavior, and tried to detect them with a good anomaly rate.

5.1.1 First Synthetic Type

The followers are consecutively following the target and all the following users have a creation time that is close to each other. The rule for the first synthetic creation time is as follows:

- Minimum Possible Creation Time can be Minimum Lower Bound of Data set.
- Maximum Possible Creation Time can not exceed the Upper Bound of the next natural follower.

The first synthetic type has two parameters: Variance of creation times and the number of synthetic data to be added.

- Variance: [1 month, 6 months, 1 year]
- Number of synthetic data: [50,250,500]

We generate a random creation time not larger than the given variance. There is an upper bound which is the follower's creation times that come after these synthetic followers. The lower bound is the minimum creation time of the data set. Other features are created after the generation of synthetic data, so we did not make any assumptions about other features. A number of total different scenarios are a combination of different parameters. In this case, we have 9 different scenarios.

Figure 5.1 shows an example of how the first synthetic type shows in a given target user. In the below example, the synthetic type has 250 synthetic users with a variance of 1 year. This shows an anomaly ratio of 3% for this given user. If we increase variance, then the first synthetic type will have a bigger difference in its creation times. It will also affect other features that use creation time, and estimated follow time. If we increase the number of synthetic data to be added, it will increase the anomaly ratio of the data set. Other parameter configurations of this user can be found in A.1 and A.2.

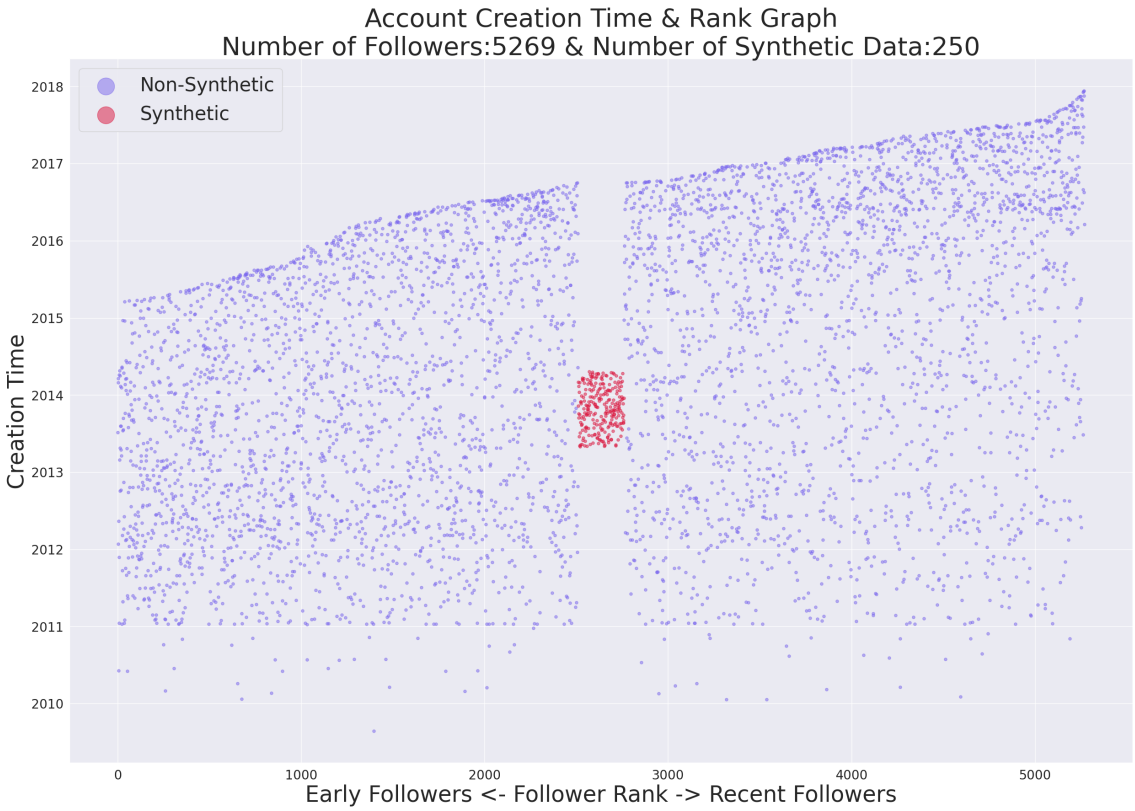

Figure 5.1 First Synthetic Type Creation Time - Rank Figure

In Figure 5.2, we observe the UMAP representation of the target shown in figure 5.1. This example has 1 month as a variance between synthetic data points. Orange points indicate the first synthetic type group. We clearly see the clusters between most of the homogenous group and the synthetic data we generated. However, we also observe another cluster. That means they are also grouped as a different cluster just like we added as anomalies. UMAP is used for representing the feature matrix in the 2D matrix for visualization. There are different configurations of parameters that result in variants of UMAPs.

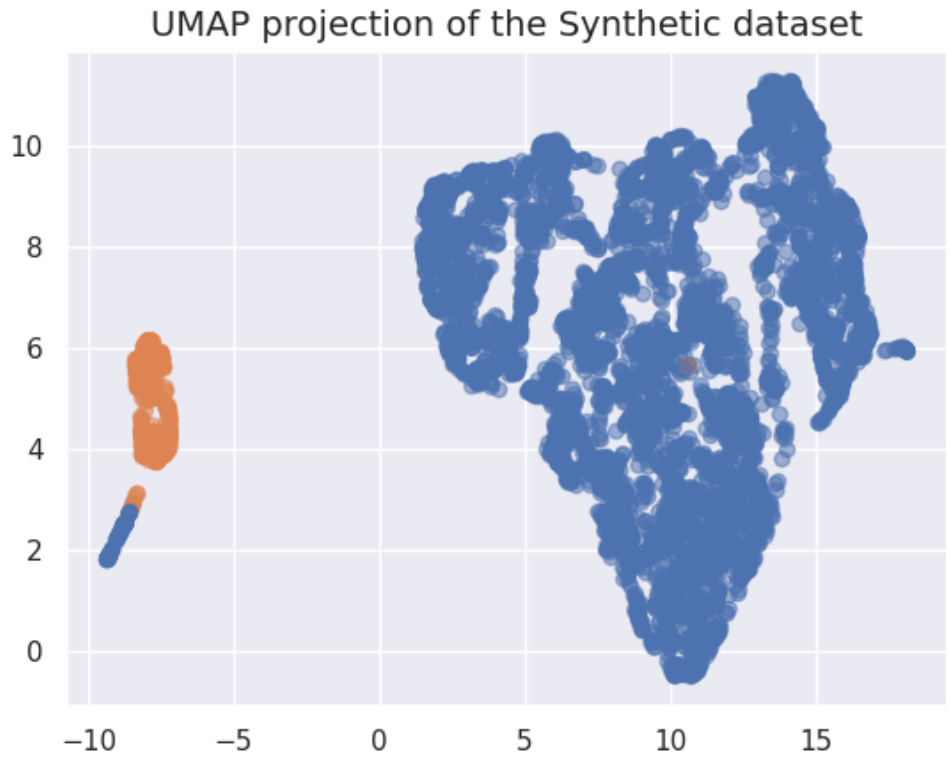

Figure 5.2 First Synthetic Type UMAP Figure

5.1.2 Second Synthetic Type

Second synthetic type is defined as the most recent followers following the target with periodically increasing creation time. Creation time ends with the maximum creation time of the data set and started to decrease periodically until the number of synthetic data N is reached. There is a second parameter called frequency. It indicates the replication of how many data points will be created. For an example scenario of, N=10 and Frequency=5: It means N multiplicate with Frequency times synthetics will be generated in the Upper Bounds of the last followers. We are taking the last 10 follower's upper bounds and creation times to generate the 50 synthetic accounts. The rule for the second synthetic creation time is as follows:

- Creation Times are always lying on the Upper Bound of Followers

The second synthetic time has 2 parameters:

- Last N followers = [10,50,100] and [5,25,50]

- Frequency = 5 and 10

The total number of second synthetic data generated is the multiplication of the last N followers and frequency. Second synthetic type scenarios are divided into two parts.

- First part: Last N followers: [10,50,100] with Frequency: 5
- Second part: Last N followers: [5,25,50] with Frequency: 10

The total number of second synthetic types is the multiplication of these parameters. These two scenarios will end up creating 50,250 and 500 synthetic types as same as the first synthetic type creations. Therefore anomaly ratios will be kept the same.

In Figure 5.3 we see the second synthetic type indicated as red points in an example target user. The synthetic data position is at the last position when we ordered the followers by rank. We are observing red points in the Upper Bounds of creation times. That is because these users try to mimic an anomaly behavior indicating they are just created recently and followed that target consecutively. In this example, the last N=50 followers are having this behavior with a frequency of 5. Since we generated Upper Bound followers at the end of the data set, we are observing a small proportion of point anomalies below the synthetic type we added. We did not label them as synthetics, however, these points can also be labeled as an anomaly in another study. Other parameter configurations of this user can be found in A.3 and A.4.

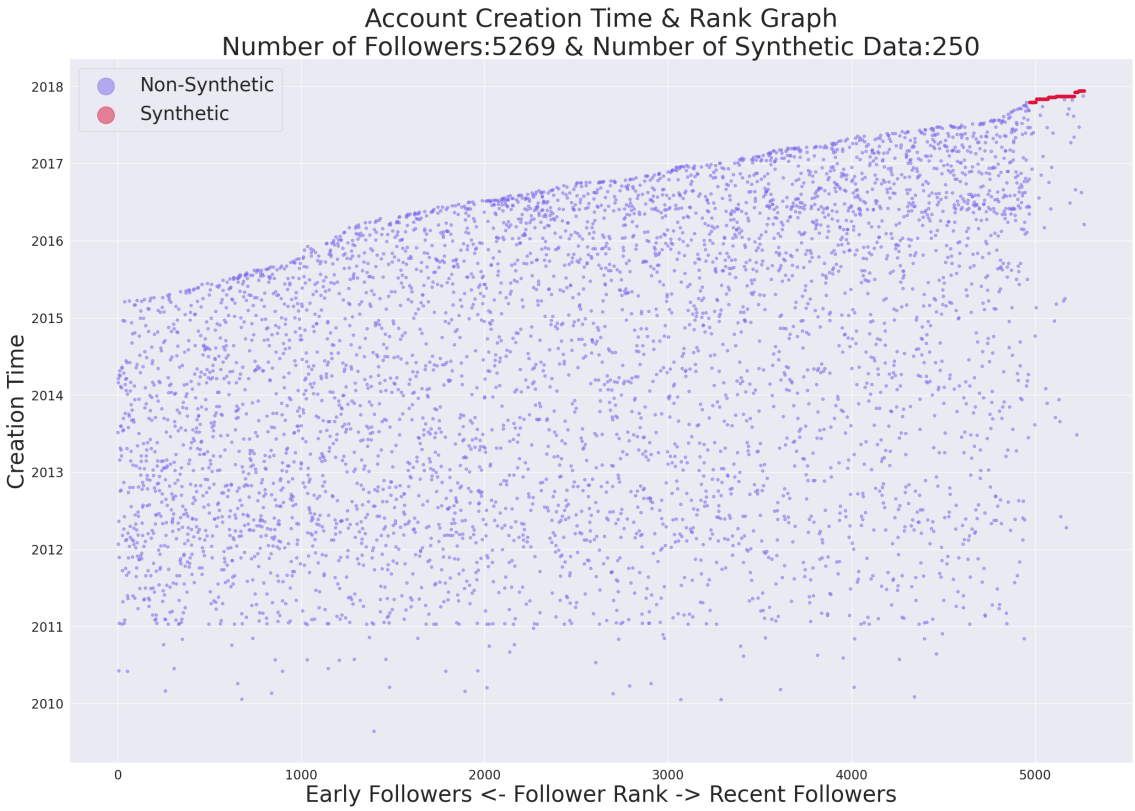

Figure 5.3 Second Synthetic Type Creation Time - Rank M=5,N=50 Figure

We see a good grouping between synthetics and homogenous data in UMAP. The feature matrix in the 2D matrix has clustered the data correctly in Figure 5.4.

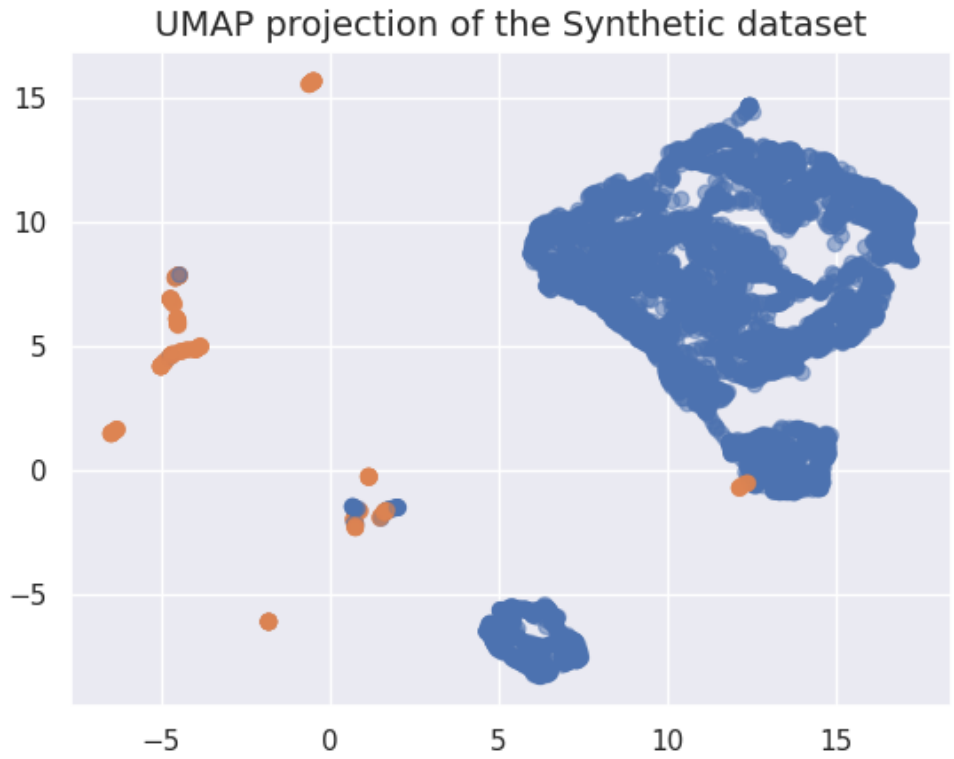

Figure 5.4 Second Synthetic Type UMAP Figure

5.1.3 Combined Synthetic Type

The last type is the combination of the first and second synthetic types. Therefore a number of complexity and parameters are merged. The number of different use cases is 18 for the combined synthetic type. We merged the parameters where the number of synthetic accounts will be the same for both the first and second synthetic types. The number of synthetics in total will be 100,500 and 1000.

In figure 5.5, we observe both synthetic types simultaneously. These synthetics will have different effects on every different target user. It is crucial to understand in figure 5.5 only shows 2 feature representations, creation time & rank. We are using 8 more features to distinguish whether these synthetics can be identified as anomalies or not. The parameters used in these figures are like below:

- Variance is 1 year with the first synthetic to be added being 500.
- Last N followers for the second synthetic type is 100 with a frequency is 5.

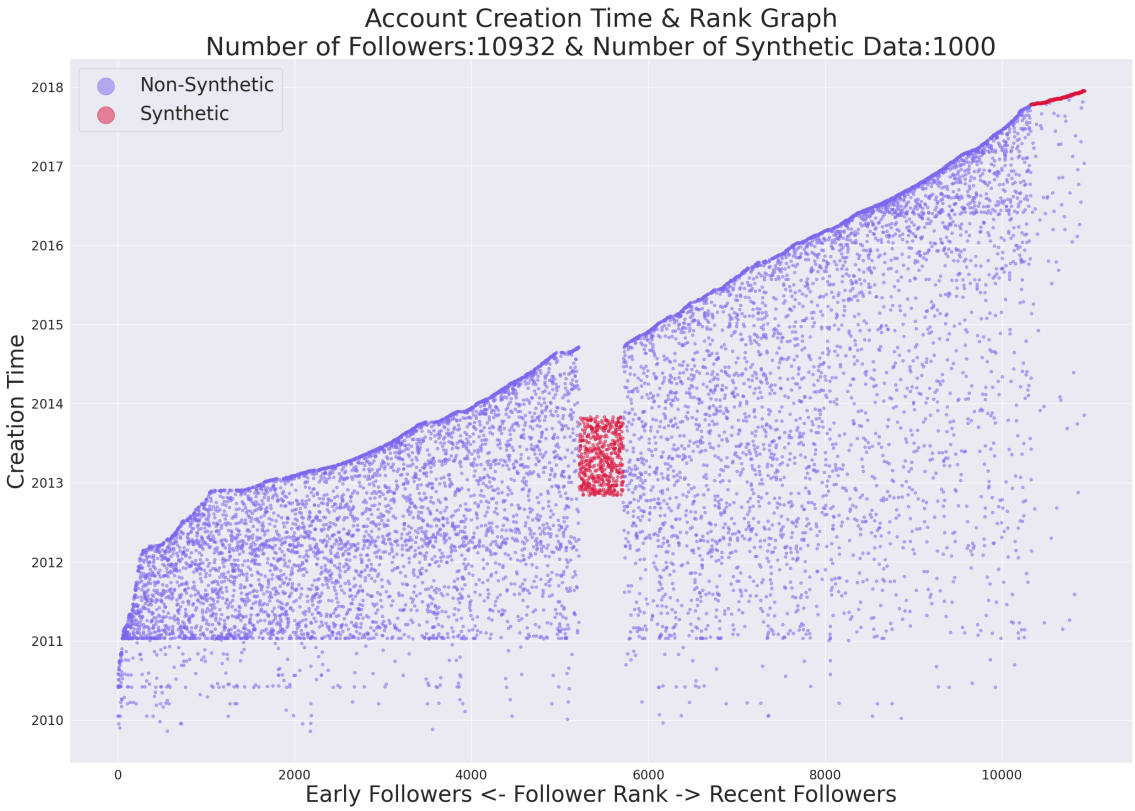

Figure 5.5 Combined Synthetic Type Creation Time - Rank Figure

5.2 Dribbble Dataset Evaluation

In order to evaluate the data, we used the Dribbble dataset which is a similar platform to Twitter discussed in the dataset section. We used logarithmic binning to randomly collect 225 targets for each bin. Bins are created as 10^3-$5*10^3$ and $5*10^3$-10^4 number of followers. There are 50 targets that are higher than the 10^4 number of followers. We also collected them to create a divergent dataset.

To make a fair evaluation, we introduced synthetic anomalies to real data we have on Dribbble accounts and tried to detect these anomalies we introduced. The challenges we faced in the project are mainly in creating the most realistic synthetic data. Right now, there are 2 types of synthetic behaviors. We also faced difficulties in creating a model that can detect synthetic anomalies from different numbers of followers. This problem is handled by using generalized features that are able to detect anomalies. However, we observed that our model evaluations are better if the target has a higher number of followers.

Combined Synthetic type scenarios have combinations of both 1st and 2nd synthetic type scenarios. We can observe their combinations in a general scenario. Figure 5.6 shows the relation between the synthetic ratio and evaluation metrics in both models.

This scenario has 1000 synthetics from 500 each synthetic types. Supervised models have higher ROC and NDCG@1000 scores between the synthetic ratios 10%-25% boundaries. Unsupervised models have a negative trend between synthetic ratio and ROC score. The good score values are between the range of 0 and 0.10.

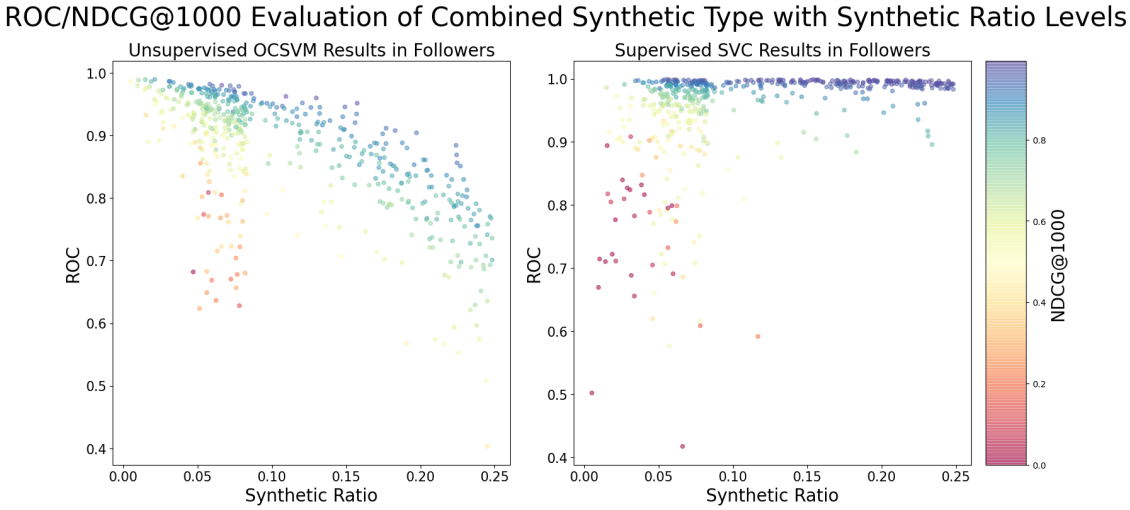

Figure 5.6 Combined Synthetic Type NDCG@1000 & ROC Supervised Scatter Figure

We also grouped target users into two categories: Low number of followers who have follower sizes between 1,000 and 5,000. And a High number of followers with sizes between 5,000-10,000 followers. We can see the changes in two categories in Figure 5.7. We can state that two bins are close to each other in unsupervised Learning algorithms with an average of 68.2% and 58.5%. There is a difference of 12% in a supervised model where low-high followers have an average of 96.6% and 84.3% respectively.

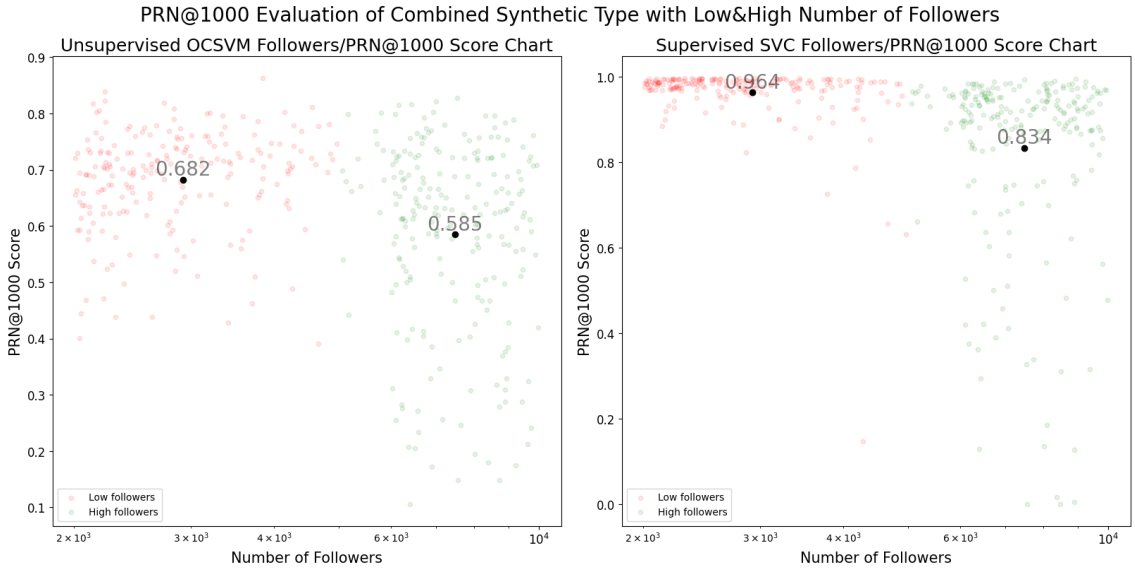

Figure 5.7 Combined Synthetic Type PRN@1000 & Number of followers Unsupervised Scatter Figure

One example of anomaly score visualization is on 5.8. Here, we see how anomaly scores are depicted on Creation Time - Rank figures. This particular instance reveals nearly 700 synthetics as the highest anomalies among followers. As depicted, the visualization also highlights various clusters as anomalies with lower scores, as well as unclassified point anomalies.

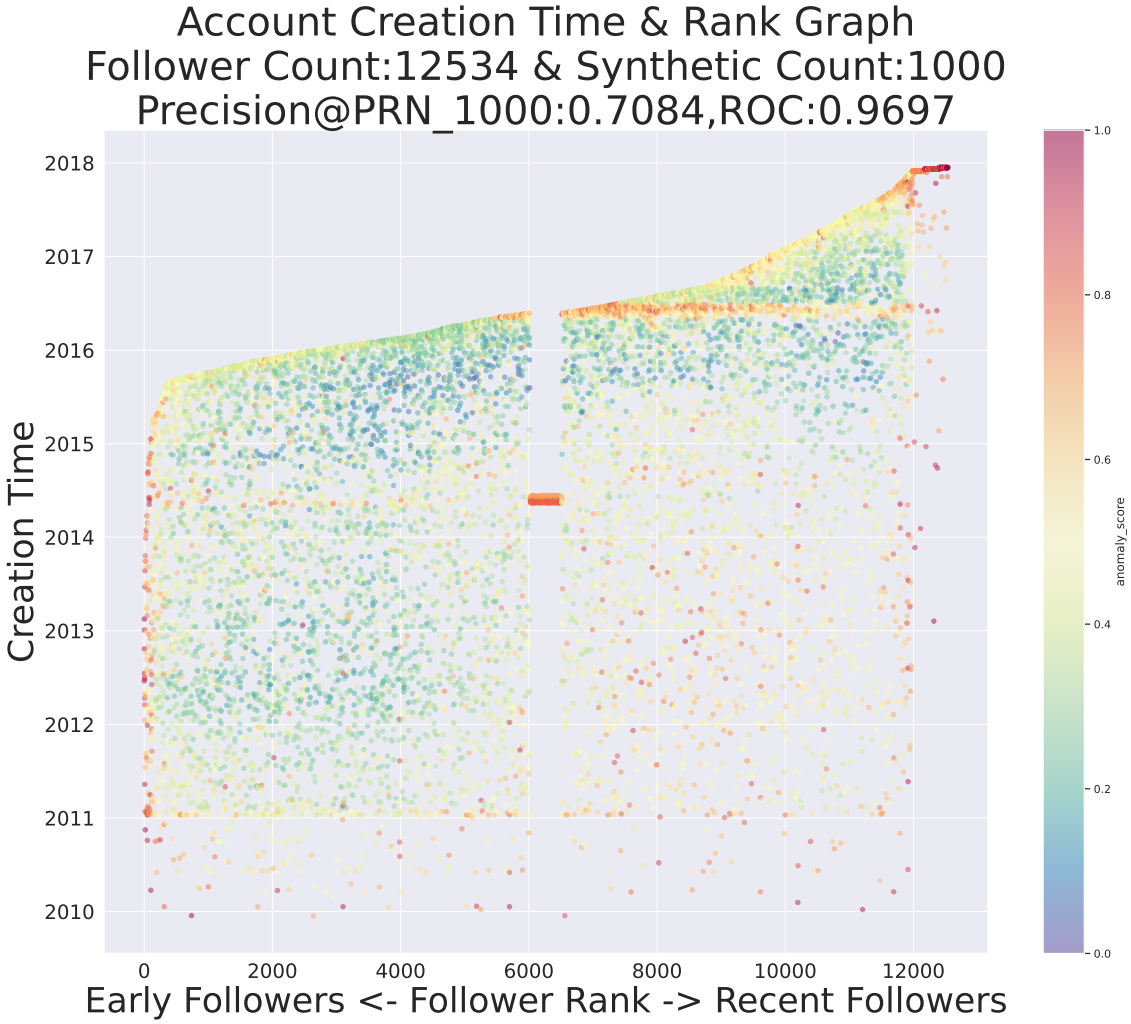

Figure 5.8 Combined Synthetic Type OCSVM Anomaly Score Scatter Figure

As displayed in 5.1, combined synthetic-type evaluation metrics results for different algorithms. Unsupervised algorithms used FeatureBagging before training and testing. We also added a two-staged anomaly model where DBSCAN first detects the major cluster. Then, the OCSVM algorithm is used for training the major cluster and predicts anomaly scores for all data points. This model is referred to as DBSCAN in our result tables. For the unsupervised learning OCSVM approach, we can state that it has better performance than all other unsupervised learning algorithms also with the highest SD. Overall, both supervised SVC & RandomForest have the highest score among all other algorithms. Other results of all scenarios are under

Appendix A.

For the first set of scenario experiments, table A.1, A.2, and A.3 present the results. The ECOD algorithm exhibits poor performance in first-scenario-type experiments. The OCSVM algorithm, on the other hand, showcases the highest performance among unsupervised models. The RandomForest algorithm slightly outperforms SVC.

We see that both unsupervised and supervised models are slightly better at detecting the first synthetic anomaly type. However, for adding 50 and 250 synthetic scenarios, supervised models are better at detecting the second anomaly type.

For the second set of scenario experiments, table A.4, A.5, and A.6 shows all the results. The OCSVM algorithm once again demonstrates superior performance among unsupervised models. Both the SVC and RandomForest algorithms exhibit impressive results. Other combined scenario experiments are A.7 and A.8. OCSVM is better overall for unsupervised, SVC is better than RandomForest for supervised.

Table 5.2 has the same scenario with added combined synthetic types, with additional 3 features. We added a number of likes, followers, and followings of each target user as an extra feature. For synthetics, our approach was to select these features randomly from the previous month's creation time bins of users. For example, if a synthetic user was created in 2020-March, its extra features will be randomly selected from real users that created their accounts in 2020-February. Though this approach is relatively flexible, it affects the performance slightly negatively compared to 5.1. If a more greedy approach is selected for the extra features, it may lead to an increase in performance.

Table 5.1 Model Results with Combined Scenario, Number of Synthetic Data:1000 Unsupervised: U, Supervised: S

					ROC		NDCG@1000		PRN@1000	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	500	50	10	0.77	0.25	0.58	0.26	0.62	0.29
			100	5	0.76	0.24	0.53	0.22	0.61	0.28
	6 months	500	50	10	0.64	0.23	0.42	0.17	0.59	0.31
			100	5	0.62	0.25	0.31	0.17	0.35	0.28
	1 yaer	500	50	10	0.70	0.12	0.49	0.13	0.74	0.26
			100	5	0.69	0.13	0.40	0.15	0.49	0.26
ECOD (U)	1 month	500	50	10	0.70	0.20	0.42	0.08	0.49	0.14
			100	5	0.72	0.19	0.39	0.08	0.40	0.10
	6 months	500	50	10	0.64	0.16	0.45	0.06	0.54	0.13
			100	5	0.65	0.16	0.40	0.07	0.45	0.11
	1 year	500	50	10	0.61	0.15	0.45	0.06	0.55	0.13
			100	5	0.61	0.15	0.41	0.07	0.46	0.10
HDBSCAN (U)	1 month	500	50	10	0.53	0.06	0.39	0.07	0.60	0.10
			100	5	0.53	0.06	0.38	0.08	0.55	0.11
	6 months	500	50	10	0.57	0.06	0.45	0.06	0.66	0.10
			100	5	0.57	0.06	0.44	0.07	0.62	0.11
	1 year	500	50	10	0.57	0.06	0.44	0.07	0.63	0.10
			100	5	0.57	0.06	0.43	0.08	0.60	0.11
IForest (U)	1 month	500	50	10	0.65	0.22	0.32	0.09	0.30	0.09
			100	5	0.67	0.21	0.32	0.10	0.27	0.09
	6 months	500	50	10	0.55	0.18	0.36	0.09	0.36	0.12
			100	5	0.55	0.17	0.35	0.10	0.32	0.11
	1 year	500	50	10	0.55	0.17	0.37	0.10	0.37	0.12
			100	5	0.54	0.16	0.36	0.10	0.34	0.11
OCSVM (U)	1 month	500	50	10	0.96	0.04	0.81	0.09	0.87	0.10
			100	5	0.92	0.07	0.74	0.10	0.82	0.12
	6 months	500	50	10	0.92	0.06	0.71	0.15	0.79	0.14
			100	5	0.89	0.08	0.63	0.16	0.65	0.17
	1 year	500	50	10	0.85	0.09	0.63	0.13	0.77	0.14
			100	5	0.83	0.10	0.55	0.15	0.60	0.16
RandomForest (S)	1 month	500	50	10	0.97	0.05	0.90	0.12	0.96	0.09
			100	5	0.96	0.06	0.83	0.18	0.91	0.16
	6 months	500	50	10	0.91	0.11	0.77	0.20	0.89	0.18
			100	5	0.94	0.08	0.73	0.24	0.81	0.24
	1 year	500	50	10	0.83	0.11	0.69	0.19	0.83	0.21
			100	5	0.88	0.13	0.64	0.29	0.72	0.29
SVC (S)	1 month	500	50	10	0.99	0.03	0.90	0.16	0.94	0.14
			100	5	0.98	0.04	0.87	0.18	0.93	0.16
	6 months	500	50	10	0.92	0.09	0.81	0.21	0.91	0.19
			100	5	0.94	0.09	0.75	0.26	0.83	0.26
	1 year	500	50	10	0.92	0.14	0.79	0.24	0.87	0.22
			100	5	0.88	0.13	0.62	0.30	0.69	0.33

Table 5.2 Model Results with Combined Scenario & Extra Features, Number of Synthetic Data:1000
Unsupervised: U, Supervised: S

					ROC		NDCG@1000		PRN@1000	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	500	50	10	0.65	0.20	0.26	0.13	0.22	0.13
			100	5	0.68	0.16	0.29	0.12	0.24	0.11
	6 months	500	50	10	0.71	0.17	0.36	0.11	0.35	0.18
			100	5	0.72	0.15	0.34	0.09	0.30	0.09
	1 year	500	50	10	0.74	0.12	0.43	0.12	0.43	0.19
			100	5	0.73	0.12	0.38	0.11	0.35	0.11
ECOD (U)	1 month	500	50	10	0.66	0.18	0.38	0.06	0.40	0.10
			100	5	0.67	0.18	0.34	0.06	0.34	0.07
	6 months	500	50	10	0.61	0.17	0.40	0.06	0.43	0.10
			100	5	0.61	0.16	0.36	0.06	0.37	0.08
	1 year	500	50	10	0.59	0.16	0.41	0.06	0.44	0.10
			100	5	0.59	0.15	0.37	0.06	0.38	0.08
HDBSCAN (U)	1 month	500	50	10	0.59	0.07	0.42	0.09	0.57	0.14
			100	5	0.59	0.07	0.40	0.10	0.51	0.14
	6 months	500	50	10	0.59	0.06	0.41	0.10	0.52	0.14
			100	5	0.59	0.06	0.39	0.11	0.46	0.15
	1 year	500	50	10	0.59	0.06	0.39	0.11	0.50	0.15
			100	5	0.59	0.07	0.37	0.11	0.43	0.15
IForest (U)	1 month	500	50	10	0.61	0.21	0.20	0.10	0.17	0.09
			100	5	0.62	0.20	0.20	0.10	0.17	0.09
	6 months	500	50	10	0.55	0.19	0.22	0.10	0.18	0.09
			100	5	0.54	0.18	0.21	0.10	0.18	0.09
	1 year	500	50	10	0.55	0.18	0.23	0.10	0.20	0.10
			100	5	0.54	0.18	0.22	0.10	0.19	0.09
OCSVM (U)	1 month	500	50	10	0.88	0.07	0.59	0.11	0.55	0.11
			100	5	0.84	0.08	0.52	0.11	0.49	0.12
	6 months	500	50	10	0.83	0.09	0.51	0.11	0.50	0.11
			100	5	0.79	0.10	0.45	0.13	0.42	0.12
	1 year	500	50	10	0.79	0.10	0.49	0.11	0.50	0.10
			100	5	0.75	0.12	0.42	0.11	0.40	0.11
RandomForest (S)	1 month	500	50	10	0.95	0.07	0.88	0.12	0.96	0.08
			100	5	0.97	0.05	0.84	0.20	0.90	0.19
	6 months	500	50	10	0.95	0.06	0.76	0.23	0.86	0.20
			100	5	0.94	0.07	0.73	0.23	0.85	0.22
	1 year	500	50	10	0.93	0.07	0.78	0.18	0.89	0.18
			100	5	0.95	0.06	0.76	0.22	0.83	0.22
SVC (S)	1 month	500	50	10	0.99	0.03	0.92	0.13	0.96	0.12
			100	5	0.99	0.03	0.93	0.12	0.96	0.10
	6 months	500	50	10	0.99	0.02	0.88	0.16	0.93	0.13
			100	5	0.95	0.06	0.78	0.23	0.86	0.21
	1 year	500	50	10	0.94	0.08	0.77	0.23	0.86	0.23
			100	5	0.94	0.09	0.75	0.26	0.81	0.26

5.3 Real-World Use Cases

We used our model in Twitter Turkish Election dataset (Najafi, Mugurtay, Demirci, Demirkiran, Karadeniz & Varol, 2022) to study the networks of politicians and their followers. Politicians' networks and all other follower information that is needed for our study are provided in this article. We selected 12 politicians from different parties and run our unsupervised models for each target's followers.

We tackled this problem in two ways. Firstly, we applied our proposed approach utilizing the features discussed in our feature extraction process. Secondly, we utilized the additional features provided in the Election dataset, which resulted in our model detecting more anomalies in the targets. We also investigated the correlation between bot scores and anomaly scores of followers utilizing the BotometerLite (Yang, Varol, Hui & Menczer, 2020). bot scores provided in the dataset, uncovering positive weak, and moderate correlations among the followers of the selected politicians.

Figure 5.9 shows a target politician's followers on Creation Time-Rank on anomaly & bot score distribution. Creation time is binned as the format of corresponding years with months, whereas the rank is binned with a thousand range. The rightmost heatmap shows where anomaly & bot scores are more frequent on the heatmap grids where the color is turned from white to black. Other politician's figures are in appendix A.5.

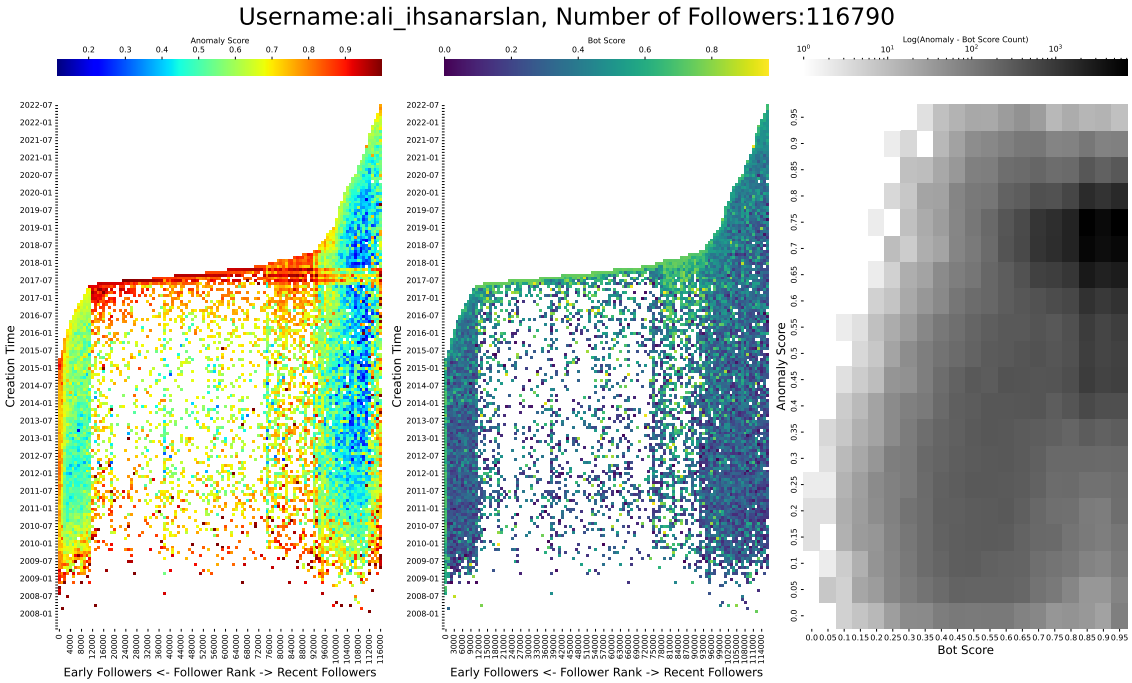

Figure 5.9 Binned Creation Time - Rank Heatmap Figure

Table 5.3 Anomaly-Bot Score Pearson Correlation Score Table

Politician Username	FC	Pearson Corr.	Pearson Corr. Extra	Spearman Corr. Extra
ali_ihsanarslan	117,442	0.27	0.563	0.426
avmustafakose	186,973	0.208	0.446	0.316
selimtemurci	624,333	0.10	0.40	0.227
mnedimyamali	194,404	0.122	0.392	0.114
dbdevletbahceli	5,530,070	0.20	0.38	0.364
vekilince	6,057,318	0.18	0.37	0.378
murat_alparslan	109,167	0.04	0.343	0.204
tahir_akyurek	149,584	0.122	0.306	0.225
m_akaydin	150,866	0.126	0.304	0.21
akadiryuksel27	20,121	0	0.123	0.051
drmusbaloglu	35,426	-0.306	-0.127	-0.126
cccemalcetin	37,661	-0.33	0.016	-0.006

Table 5.3 shows the correlation between anomaly score & bot score. We created two experiments with our features and extra features like verified, status count, followers count, default profile image, favorites count, and friends count. Extra features have a high positive effect on the correlation between the two scores. FC column in the table stands for the Follower Count of the target politician. We divide the politicians horizontally who have less than 100,000 followers. We can state that there is a moderate correlation when the number of followers who followed the politician is higher than 100,000 users.

We also compared Botometer's time performance with our model. Botometer fetches data using Twitter API. That is why they are restricted by the rate limits of 180 requests per 15 minutes. We take one example account having 100 users. It took Botometer to give a bot score for every follower for around 15 minutes. Our model runtime was less than 1 second because we did not have rate limit restriction and used the collected election dataset. However, this number increases when the target user has more followers. Runtime increases exponentially when the number of followers of an account increases.

6. CONCLUSION

In this thesis, we proposed supervised and unsupervised learning methods for the task of detecting anomalies in selected social media platforms. Through a series of carefully crafted experiments, we demonstrate the feasibility of our approach using social data. Our methodology is enriched by the inclusion of various features, such as the rank order and creation time of followers, which imbue our methods with a powerful ability to uncover hidden patterns in the data. We validate the effectiveness of our approach using the Dribbble dataset, measuring its performance using standard metrics such as NDCG, Precision, and ROC scores.

Our results show that unsupervised methods are particularly adept at detecting both experimental and synthetic scenarios, while supervised methods exhibit impressive overall performance. However, it must be noted that in the realm of real-world datasets, supervised methods may be susceptible to bias in their ability to capture anomalies.

This study makes an improvement in bot score literature by implementing faster and easier models rather than higher complexity of bot detection models. Overall, this thesis represents a significant contribution to the field of anomaly detection in social media platforms.

6.0.1 Research Questions

In order to conclude, we need to emphasize the answers to research questions with our findings:

- RQ1) Which features will be included in the model, do the chosen features represent real-world scenarios?

  A1) This is stated in the follower feature extraction section 4.2. We proposed 10 powerful features. Two features come from the API, creation time and

rank. Other features are generated by using two features. Our focus is on leveraging the inherent neighbor information, boundary estimates of following times, and the relative rank of each follower to generate features that are capable of representing the complexities of real-world scenarios. The efficacy of our approach is demonstrated through a series of rigorous experiments, which showcase its impressive detection rate and overall performance.

- RQ2) How to select the algorithms used in classifying?

  A2) We selected algorithms that are robust, fast, and give an overall good score from our evaluation metrics.

- RQ3) How effective is the model selected in the evaluation phase?

  A3) We measure the effectiveness of all the models selected in 5.1. We can state models are affected if they are overall higher results that are above 50% in all three evaluation metrics.

6.0.2 Limitations

It is crucial to know the limitations of our models. It is hard to identify which accounts are anomalous and inorganic during special events. For example, when a public figure makes a speech, inorganic follower activities may rise in social networks. These activities may be coordinated and have a purpose behind them. But these activities may also be due to the special event effect. Follower activities may have the illusion to be inorganic accounts when in fact they are simply natural followers that appeared to be anomalous. Thus it is crucial to identify the mechanism behind a fake follower with a natural follower knowing the limitations.

6.0.3 Future Work

Future work of this study can be held in two ways. One way is to enhance the architecture by selecting new features, and models that potentially have better evaluation metric scores. Importantly, it will be crucial to maintain the simplicity of the current model in any modifications. We also created experiments for additional features such as the number of followers, followings, and likes. Other models may

focus on more text data features such as tweets, and hashtags.

Secondly, we can explore case studies to highlight the capabilities of the proposed methods in real-world scenarios. For instance, we examined a sample case of the impact of anomalies on social media during the election process. Additionally, we can investigate the effects of anomalous behavior in the influencer market. We can examine more case studies in real-world scenarios.

BIBLIOGRAPHY

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, 90–113.

Anand, A., Dutta, S., & Mukherjee, P. (2020). Influencer marketing with fake followers. *IIM Bangalore Research Paper*, (580).

Anantharam, P., Thirunarayan, K., & Sheth, A. (2012). Topical anomaly detection from twitter stream. In *Proceedings of the 4th Annual ACM Web Science Conference*, (pp. 11–14).

Bello, B. S., Heckel, R., & Minku, L. (2018). Reverse engineering the behaviour of twitter bots. In *2018 Fifth International Conference on social networks analysis, management and security (SNAMS)*, (pp. 27–34). IEEE.

Bruno, M., Lambiotte, R., & Saracco, F. (2022). Brexit and bots: characterizing the behaviour of automated accounts on twitter during the uk election. *EPJ Data Science*, *11*(1), 17.

Campello, R. J., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *10*(1), 1–51.

Carmona, C. U., Aubet, F.-X., Flunkert, V., & Gasthaus, J. (2021). Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1–58.

Chang, C.-I. & Chiang, S.-S. (2002). Anomaly detection and classification for hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, *40*(6), 1314–1325.

Chen, X., Gao, S., & Zhang, X. (2021). Visual analysis of global research trends in social bots based on bibliometrics. *Online Information Review*.

Confessore, N., Dance, G. J., Harris, R., & Hansen, M. (2018). The follower factory. *The New York Times*, *27*.

da Costa, K. A., Papa, J. P., Passos, L. A., Colombo, D., Del Ser, J., Muhammad, K., & de Albuquerque, V. H. C. (2020). A critical literature survey and prospects on tampering and anomaly detection in image data. *Applied Soft Computing*, *97*, 106727.

Domínguez, D. R., Redondo, R. P. D., Vilas, A. F., & Khalifa, M. B. (2017). Sensing the city with instagram: Clustering geolocated data for outlier detection. *Expert systems with applications*, *78*, 319–333.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, *59*(7), 96–104.

Fisch, A. T., Eckley, I. A., & Fearnhead, P. (2022). Subset multivariate collective and point anomaly detection. *Journal of Computational and Graphical Statistics*, *31*(2), 574–585.

Garroppo, R. G. & Niccolini, S. (2018). Anomaly detection mechanisms to find social events using cellular traffic data. *Computer Communications*, *116*, 240–252.

Hao, Y., Li, J., Wang, N., Wang, X., & Gao, X. (2022). Spatiotemporal consistency-enhanced network for video anomaly detection. *Pattern Recognition*, *121*,

108232.

Hendrycks, D., Mazeika, M., & Dietterich, T. (2018). Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Ji, Y. & Lee, H. (2022). Event-based anomaly detection using a one-class svm for a hybrid electric vehicle. *IEEE Transactions on Vehicular Technology*, *71*(6), 6032–6043.

Kumar, S., Khan, M. B., Hasanat, M. H. A., Saudagar, A. K. J., AlTameem, A., & AlKhathami, M. (2022). An anomaly detection framework for twitter data. *Applied Sciences*, *12*(21), 11059.

Kyriienko, O. & Magnusson, E. B. (2022). Unsupervised quantum machine learning for fraud detection. *arXiv preprint arXiv:2208.01203*.

Lee, S. & Kim, J. (2014). Early filtering of ephemeral malicious accounts on twitter. *Computer communications*, *54*, 48–57.

Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., & Chen, G. (2022). Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*.

Mahapatra, A., Srivastava, N., & Srivastava, J. (2012). Contextual anomaly detection in text data. *Algorithms*, *5*(4), 469–489.

McInnes, L. & Healy, J. (2017). Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, (pp. 33–42). IEEE.

Meeder, B. (2011). We know who you followed last summer: inferring social link creation times in twitter. *ACM Digital Library*, 517–526.

Najafi, A., Mugurtay, N., Demirci, E., Demirkiran, S., Karadeniz, H. A., & Varol, O. (2022). # secim2023: First public dataset for studying turkish general election. *arXiv preprint arXiv:2211.13121*.

Omar, S., Ngadi, A., & Jebur, H. H. (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, *79*(2).

Patrikar, D. R. & Parate, M. R. (2022). Anomaly detection using edge computing in video surveillance system. *International Journal of Multimedia Information Retrieval*, 1–26.

Ruiz Soler, J. (2017). Twitter research for social scientists: A brief introduction to the benefits, limitations and tools for analysing twitter data.

Safety, T. (2008). Four truths about bots. `https://blog.twitter.com/common-thread/en/topics/stories/2021/four-truths-about-bots/`. [Online; accessed 21-September-2021].

Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, *96*, 104.

Shao, Y., Zhang, W., Liu, P., Huyue, R., Tang, R., Yin, Q., & Li, Q. (2022). Log anomaly detection method based on bert model optimization. In *2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, (pp. 161–166). IEEE.

Staff (2022). Alexa Prize TaskBot Challenge. `https://www.amazon.science/alexa-prize/socialbot-grand-challenge/2022/`. [Online; accessed 28-June-2022].

Thi, N. N., Cao, V. L., & Le-Khac, N.-A. (2018). One-class collective anomaly detection based on long short-term memory recurrent neural networks. *arXiv*

preprint arXiv:1802.00324.

Ullah, W., Ullah, A., Hussain, T., Muhammad, K., Heidari, A. A., Del Ser, J., Baik, S. W., & De Albuquerque, V. H. C. (2022). Artificial intelligence of things-assisted two-stream neural network for anomaly detection in surveillance big video data. *Future Generation Computer Systems*, *129*, 286–297.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, (pp. 1–5). IEEE.

Varol, O., Davis, C. A., Menczer, F., & Flammini, A. (2018). Feature engineering for social bot detection. In *Feature engineering for machine learning and data analytics* (pp. 311–334). CRC Press.

Varol, O. & Uluturk, I. (2020). Journalists on twitter: self-branding, audiences, and involvement of bots. *Journal of Computational Social Science*, *3*(1), 83–101.

Wang, X., Varol, O., & Eliassi-Rad, T. (2021). Information access equality on network generative models. *arXiv preprint arXiv:2107.02263*.

Weng, L., Ratkiewicz, J., Perra, N., Gonçalves, B., Castillo, C., Bonchi, F., Schifanella, R., Menczer, F., & Flammini, A. (2013). The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 356–364).

Yang, K.-C., Varol, O., Hui, P.-M., & Menczer, F. (2020). Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, (pp. 1096–1103).

Zhang, M., Qi, X., Chen, Z., & Liu, J. (2022). Social bots' involvement in the covid-19 vaccine discussions on twitter. *International Journal of Environmental Research and Public Health*, *19*(3), 1651.

Zimmer, M. & Proferes, N. J. (2014). A topology of twitter research: disciplines, methods, and ethics. *Aslib Journal of Information Management*.

APPENDIX A

First Synthetic Type

Example Figures for First Synthetic Type Scenario:

Configurations: Variance: 1 month, Synthetic Count:250

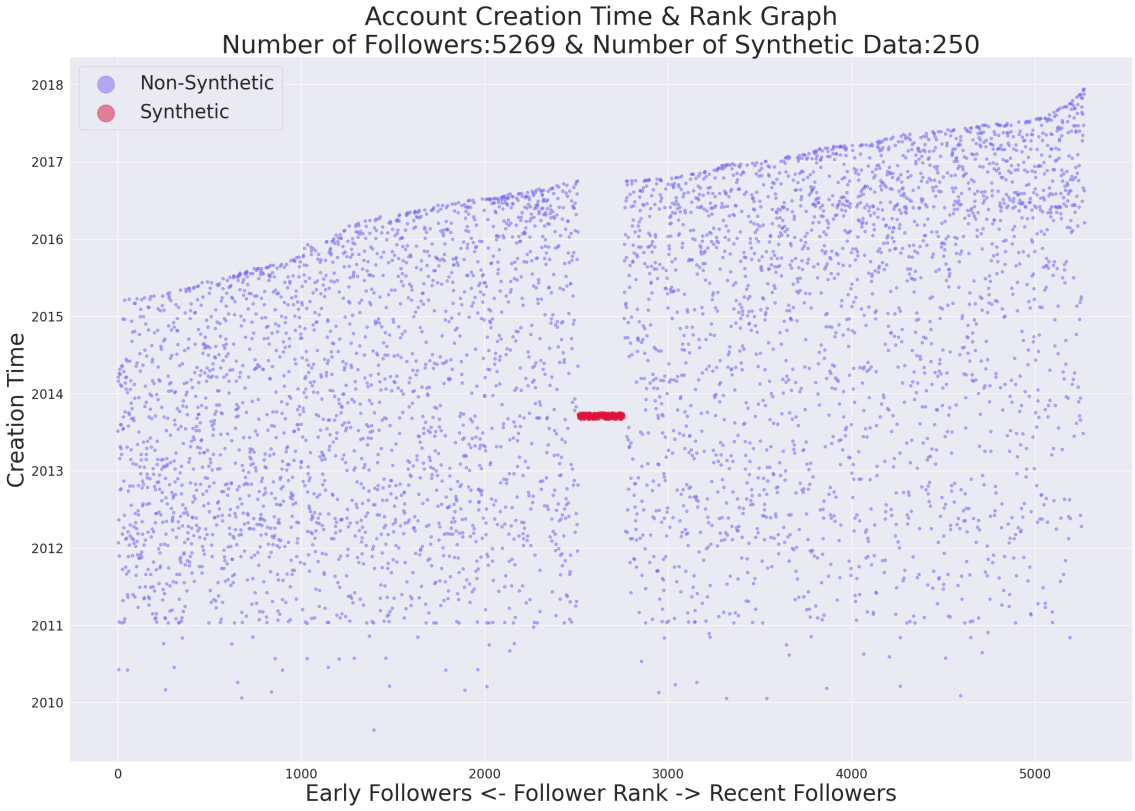

Figure A.1 First Synthetic Type Creation Time - Rank Figure

Configurations: Variance: 6 months, Synthetic Count:250

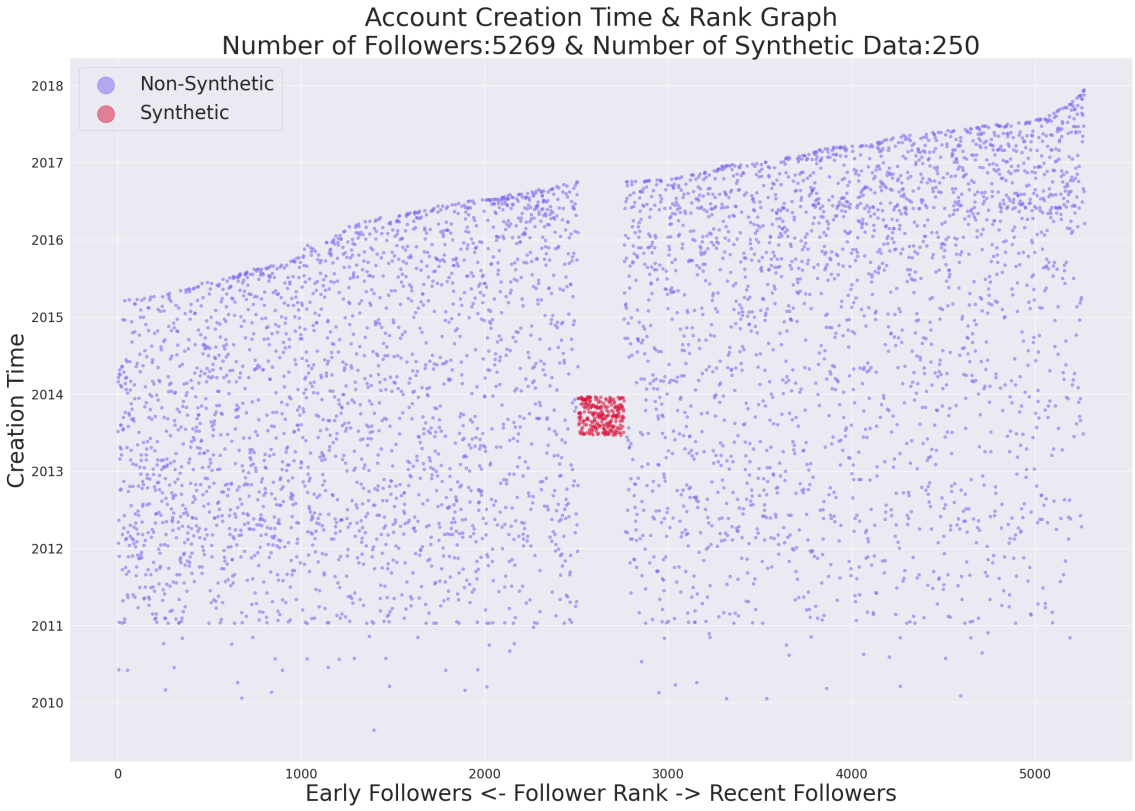

Figure A.2 First Synthetic Type Creation Time - Rank Figure

Table A.1 Model Results with First Scenario, Number of Synthetic Data:50
Unsupervised: U, Supervised: S

			ROC		NDCG@50		PRN@50	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	50	0.92	0.07	0.38	0.34	0.43	0.39
	6 months	50	0.84	0.11	0.05	0.09	0.06	0.14
	1 year	50	0.74	0.15	0.03	0.05	0.04	0.07
ECOD (U)	1 month	50	0.81	0.08	0.03	0.07	0.03	0.08
	6 months	50	0.66	0.13	0.01	0.02	0.01	0.03
	1 year	50	0.55	0.17	0.00	0.01	0.00	0.01
HDBSCAN (U)	1 month	50	0.65	0.09	0.26	0.14	0.56	0.25
	6 months	50	0.52	0.10	0.18	0.15	0.37	0.29
	1 year	50	0.58	0.11	0.13	0.15	0.21	0.24
IForest (U)	1 month	50	0.81	0.11	0.01	0.05	0.01	0.05
	6 months	50	0.68	0.15	0.00	0.04	0.01	0.05
	1 year	50	0.61	0.19	0.01	0.03	0.01	0.04
OCSVM (U)	1 month	50	0.93	0.04	0.32	0.27	0.33	0.30
	6 months	50	0.85	0.10	0.06	0.08	0.07	0.10
	1 year	50	0.74	0.15	0.03	0.05	0.04	0.08
RandomForest (S)	1 month	50	0.87	0.19	0.48	0.41	0.55	0.45
	6 months	50	0.89	0.12	0.32	0.31	0.39	0.35
	1 year	50	0.81	0.17	0.25	0.31	0.28	0.34
SVC (S)	1 month	50	0.91	0.14	0.58	0.41	0.63	0.44
	6 months	50	0.90	0.14	0.30	0.33	0.33	0.36
	1 year	50	0.78	0.24	0.28	0.35	0.30	0.38

Table A.2 Model Results with First Scenario, Number of Synthetic Data:250
Unsupervised: U, Supervised: S

			ROC		NDCG@250		PRN@250	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	250	0.94	0.09	0.58	0.30	0.58	0.32
	6 months	250	0.70	0.20	0.06	0.12	0.06	0.13
	1 year	250	0.70	0.14	0.09	0.09	0.08	0.08
ECOD (U)	1 month	250	0.77	0.12	0.05	0.07	0.04	0.06
	6 months	250	0.63	0.13	0.03	0.05	0.02	0.04
	1 year	250	0.51	0.17	0.01	0.03	0.01	0.02
HDBSCAN (U)	1 month	250	0.61	0.09	0.20	0.08	0.31	0.12
	6 months	250	0.69	0.08	0.33	0.12	0.44	0.19
	1 year	250	0.64	0.06	0.25	0.12	0.33	0.17
IForest (U)	1 month	250	0.69	0.19	0.02	0.02	0.02	0.03
	6 months	250	0.46	0.22	0.00	0.02	0.00	0.02
	1 year	250	0.39	0.24	0.00	0.02	0.00	0.02
OCSVM (U)	1 month	250	0.98	0.03	0.71	0.23	0.72	0.24
	6 months	250	0.88	0.10	0.26	0.18	0.25	0.18
	1 year	250	0.73	0.15	0.12	0.11	0.11	0.10
RandomForest (S)	1 month	250	0.94	0.13	0.78	0.31	0.82	0.32
	6 months	250	0.67	0.23	0.33	0.39	0.35	0.41
	1 year	250	0.92	0.13	0.67	0.29	0.72	0.29
SVC (S)	1 month	250	0.97	0.09	0.85	0.29	0.87	0.30
	6 months	250	0.86	0.16	0.48	0.43	0.48	0.44
	1 year	250	0.94	0.15	0.67	0.33	0.68	0.35

Table A.3 Model Results with First Scenario, Number of Synthetic Data:500
Unsupervised: U, Supervised: S

			ROC		NDCG@500		PRN@500	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	500	0.88	0.18	0.59	0.30	0.61	0.32
	6 months	500	0.56	0.26	0.04	0.09	0.03	0.08
	1 year	500	0.63	0.13	0.11	0.12	0.10	0.10
ECOD (U)	1 month	500	0.65	0.20	0.06	0.06	0.05	0.05
	6 months	500	0.54	0.17	0.04	0.05	0.03	0.04
	1 year	500	0.44	0.17	0.02	0.04	0.02	0.03
HDBSCAN (U)	1 month	500	0.57	0.10	0.22	0.08	0.27	0.10
	6 months	500	0.66	0.10	0.35	0.10	0.43	0.16
	1 year	500	0.65	0.10	0.33	0.10	0.41	0.16
IForest (U)	1 month	500	0.52	0.28	0.02	0.03	0.02	0.02
	6 months	500	0.30	0.22	0.00	0.01	0.00	0.01
	1 year	500	0.25	0.23	0.00	0.01	0.00	0.01
OCSVM (U)	1 month	500	0.98	0.03	0.79	0.17	0.82	0.18
	6 months	500	0.87	0.10	0.38	0.20	0.36	0.20
	1 year	500	0.71	0.14	0.19	0.15	0.17	0.14
RandomForest (S)	1 month	500	0.99	0.07	0.92	0.16	0.95	0.15
	6 months	500	0.70	0.22	0.24	0.31	0.27	0.34
	1 year	500	0.92	0.14	0.71	0.29	0.75	0.29
SVC (S)	1 month	500	0.98	0.03	0.80	0.29	0.82	0.26
	6 months	500	0.70	0.33	0.45	0.45	0.45	0.46
	1 year	500	0.95	0.09	0.77	0.29	0.78	0.30

Second Synthetic Type

Example Figures for Second Synthetic Type Scenario:

Configurations: M=5, N=10

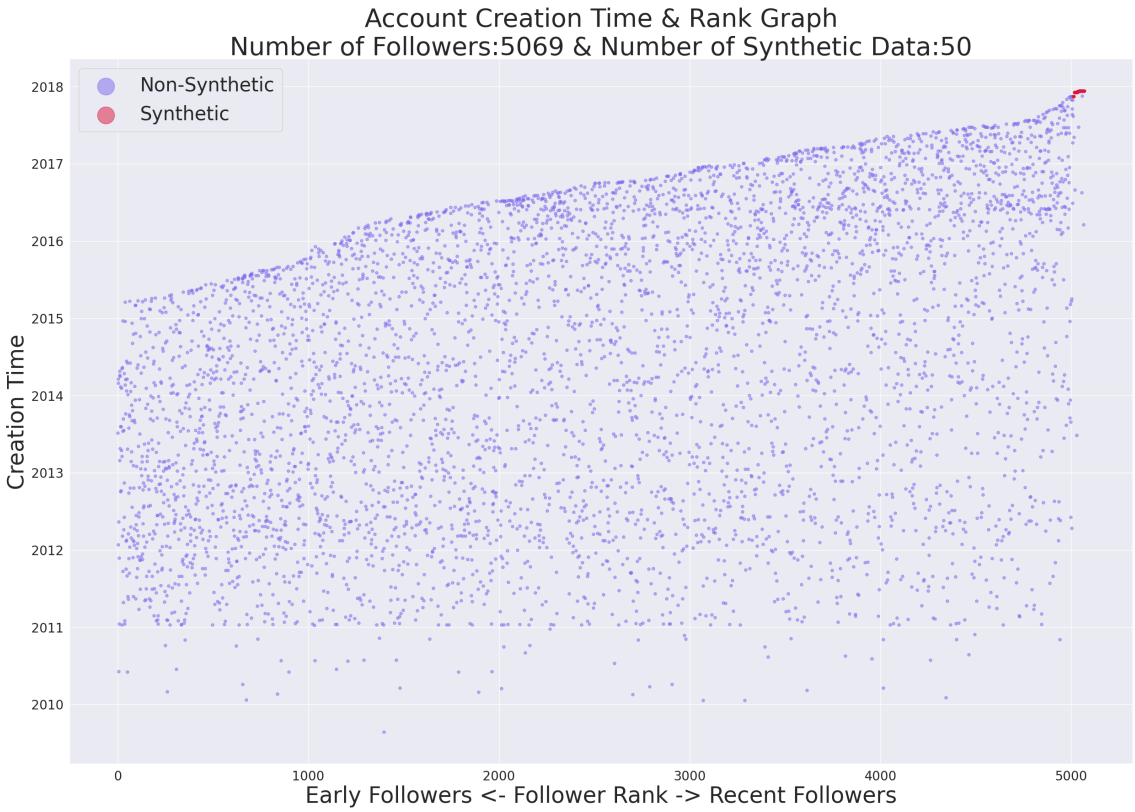

Figure A.3 First Synthetic Type M=5,N10 Figure

Configurations: M=5, N=100

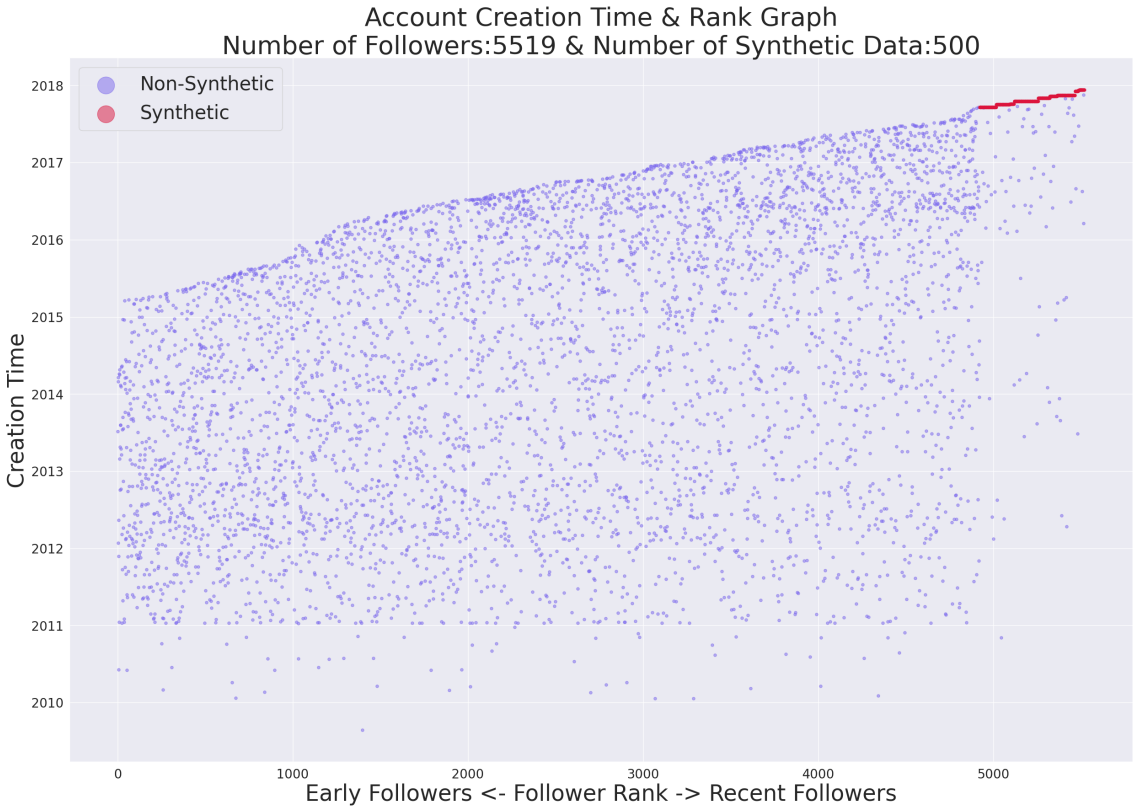

Figure A.4 Second Synthetic Type M=5,N=100 Figure

Table A.4 Model Results with Second Scenario, Number of Synthetic Data:50 Unsupervised: U, Supervised: S

			ROC		NDCG@50		PRN@50	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	5	10	0.97	0.03	0.55	0.26	0.68	0.30
	10	5	0.94	0.04	0.29	0.20	0.42	0.28
ECOD (U)	5	10	0.99	0.01	0.61	0.16	0.70	0.15
	10	5	0.99	0.01	0.48	0.16	0.55	0.17
HDBSCAN (U)	5	10	0.50	0.11	0.36	0.16	0.77	0.26
	10	5	0.47	0.10	0.29	0.16	0.62	0.31
IForest (U)	5	10	0.98	0.01	0.27	0.20	0.27	0.21
	10	5	0.97	0.02	0.21	0.17	0.21	0.18
OCSVM (U)	5	10	0.97	0.02	0.40	0.18	0.54	0.24
	10	5	0.94	0.04	0.24	0.15	0.35	0.21
RandomForest (S)	5	10	0.93	0.14	0.70	0.34	0.77	0.35
	10	5	0.95	0.06	0.32	0.41	0.34	0.43
SVC (S)	5	10	0.90	0.16	0.55	0.42	0.63	0.43
	10	5	0.90	0.14	0.46	0.41	0.52	0.44

Table A.5 Model Results with Second Scenario, Number of Synthetic Data:250 Unsupervised: U, Supervised:S

			ROC		NDCG@250		PRN@250	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	25	10	0.96	0.06	0.68	0.29	0.74	0.30
	50	5	0.92	0.07	0.45	0.27	0.50	0.30
ECOD (U)	25	10	0.96	0.04	0.58	0.12	0.62	0.14
	50	5	0.94	0.04	0.49	0.12	0.51	0.13
HDBSCAN (U)	25	10	0.49	0.05	0.37	0.09	0.70	0.15
	50	5	0.49	0.05	0.34	0.09	0.63	0.17
IForest (U)	25	10	0.94	0.06	0.34	0.18	0.32	0.18
	50	5	0.94	0.05	0.31	0.17	0.28	0.16
OCSVM (U)	25	10	0.97	0.03	0.68	0.17	0.76	0.18
	50	5	0.94	0.05	0.50	0.18	0.53	0.18
RandomForest (S)	25	10	0.99	0.02	0.91	0.19	0.93	0.18
	50	5	0.90	0.14	0.60	0.42	0.64	0.44
SVC (S)	25	10	0.99	0.07	0.86	0.26	0.87	0.26
	50	5	0.92	0.19	0.79	0.31	0.84	0.30

Table A.6 Model Results with Second Scenario, Number of Synthetic Data:500 Unsupervised: U, Supervised:S

			ROC		NDCG@500		PRN@500	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	50	10	0.93	0.10	0.73	0.25	0.78	0.27
	100	5	0.84	0.15	0.48	0.28	0.51	0.32
ECOD (U)	50	10	0.90	0.10	0.53	0.12	0.55	0.15
	100	5	0.87	0.11	0.47	0.11	0.48	0.12
HDBSCAN (U)	50	10	0.49	0.04	0.40	0.07	0.71	0.11
	100	5	0.48	0.04	0.36	0.08	0.64	0.13
IForest (U)	50	10	0.86	0.13	0.36	0.14	0.35	0.14
	100	5	0.87	0.10	0.33	0.14	0.29	0.13
OCSVM (U)	50	10	0.96	0.04	0.75	0.15	0.82	0.14
	100	5	0.91	0.08	0.58	0.17	0.60	0.18
RandomForest (S)	50	10	0.95	0.10	0.80	0.29	0.86	0.28
	100	5	0.91	0.13	0.71	0.31	0.77	0.31
SVC (S)	50	10	0.98	0.04	0.91	0.18	0.95	0.16
	100	5	0.95	0.08	0.76	0.31	0.81	0.31

Combined Synthetic Type

Example Figures for Combined Synthetic Type Scenario:

Table A.7 Model Results with Combined Scenario, Number of Synthetic Data:100 Unsupervised: U, Supervised: S

					ROC		NDCG@100		PRN@100	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	50	5	10	0.95	0.04	0.57	0.19	0.72	0.24
			10	5	0.93	0.04	0.45	0.19	0.56	0.27
	6 months	50	5	10	0.90	0.05	0.44	0.15	0.65	0.27
			10	5	0.89	0.06	0.27	0.14	0.36	0.22
	1 year	50	5	10	0.85	0.08	0.43	0.15	0.68	0.26
			10	5	0.84	0.07	0.26	0.14	0.37	0.23
ECOD (U)	1 month	50	5	10	0.88	0.04	0.47	0.09	0.66	0.14
			10	5	0.89	0.04	0.40	0.10	0.50	0.15
	6 months	50	5	10	0.81	0.07	0.47	0.09	0.67	0.14
			10	5	0.81	0.07	0.40	0.10	0.51	0.15
	1 year	50	5	10	0.75	0.09	0.47	0.09	0.67	0.14
			10	5	0.76	0.08	0.40	0.10	0.52	0.15
HDBSCAN	1 month	50	5	10	0.57	0.07	0.36	0.11	0.69	0.18
			10	5	0.56	0.07	0.32	0.12	0.62	0.20
	6 months	50	5	10	0.51	0.07	0.33	0.11	0.69	0.20
			10	5	0.50	0.07	0.29	0.12	0.59	0.23
	1 year	50	5	10	0.54	0.08	0.31	0.12	0.68	0.21
			10	5	0.53	0.07	0.27	0.14	0.55	0.25
IForest	1 month	50	5	10	0.89	0.05	0.30	0.13	0.32	0.17
			10	5	0.89	0.05	0.23	0.13	0.24	0.14
	6 months	50	5	10	0.83	0.07	0.31	0.14	0.34	0.18
			10	5	0.82	0.08	0.24	0.14	0.26	0.16
	1 year	50	5	10	0.79	0.09	0.31	0.14	0.34	0.18
			10	5	0.78	0.09	0.24	0.14	0.26	0.17
OCSVM	1 month	50	5	10	0.95	0.02	0.52	0.17	0.63	0.19
			10	5	0.94	0.03	0.42	0.18	0.49	0.22
	6 months	50	5	10	0.92	0.05	0.39	0.15	0.54	0.20
			10	5	0.90	0.06	0.27	0.14	0.33	0.17
	1 year	50	5	10	0.86	0.07	0.36	0.13	0.53	0.20
			10	5	0.84	0.07	0.24	0.13	0.32	0.17
RandomForest	1 month	50	5	10	0.90	0.12	0.69	0.23	0.86	0.21
			10	5	0.89	0.12	0.58	0.22	0.74	0.25
	6 months	50	5	10	0.92	0.09	0.59	0.26	0.75	0.27
			10	5	0.93	0.06	0.58	0.23	0.74	0.25
	1 year	50	5	10	0.84	0.10	0.55	0.19	0.79	0.24
			10	5	0.86	0.11	0.31	0.29	0.42	0.36
SVC	1 month	50	5	10	0.89	0.15	0.64	0.27	0.82	0.28
			10	5	0.87	0.11	0.59	0.17	0.79	0.16
	6 months	50	5	10	0.90	0.09	0.47	0.34	0.58	0.39
			10	5	0.96	0.04	0.57	0.22	0.69	0.28
	1 year	50	5	10	0.75	0.17	0.51	0.22	0.72	0.31
			10	5	0.80	0.15	0.40	0.31	0.51	0.36

Table A.8 Model Results with Combined Scenario, Number of Synthetic Data:500 Unsupervised: U, Supervised: S

					ROC		NDCG@500		PRN@500	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	250	25	10	0.91	0.11	0.63	0.21	0.69	0.24
			50	5	0.88	0.13	0.54	0.21	0.62	0.26
	6 months	250	25	10	0.77	0.16	0.44	0.16	0.62	0.29
			50	5	0.78	0.14	0.34	0.15	0.38	0.24
	1 year	250	25	10	0.79	0.09	0.48	0.15	0.70	0.28
			50	5	0.79	0.09	0.38	0.16	0.47	0.26
ECOD (U)	1 month	250	25	10	0.82	0.10	0.44	0.07	0.56	0.12
			50	5	0.83	0.10	0.40	0.08	0.44	0.11
	6 months	250	25	10	0.73	0.11	0.46	0.07	0.59	0.13
			50	5	0.74	0.10	0.41	0.08	0.48	0.11
	1 year	250	25	10	0.69	0.11	0.46	0.07	0.60	0.12
			50	5	0.69	0.11	0.41	0.08	0.49	0.11
HDBSCAN (U)	1 month	250	25	10	0.55	0.05	0.35	0.08	0.59	0.11
			50	5	0.55	0.06	0.33	0.08	0.53	0.13
	6 months	250	25	10	0.58	0.05	0.42	0.08	0.64	0.13
			50	5	0.59	0.05	0.41	0.09	0.60	0.15
	1 year	250	25	10	0.56	0.04	0.39	0.09	0.61	0.13
			50	5	0.57	0.04	0.37	0.09	0.56	0.14
IForest (U)	1 month	250	25	10	0.79	0.14	0.34	0.11	0.31	0.11
			50	5	0.79	0.13	0.32	0.12	0.28	0.11
	6 months	250	25	10	0.68	0.15	0.37	0.12	0.36	0.15
			50	5	0.68	0.14	0.34	0.12	0.33	0.14
	1 year	250	25	10	0.66	0.14	0.38	0.11	0.38	0.15
			50	5	0.65	0.14	0.35	0.12	0.34	0.15
OCSVM (U)	1 month	250	25	10	0.97	0.03	0.77	0.14	0.83	0.13
			50	5	0.95	0.04	0.68	0.14	0.75	0.16
	6 months	250	25	10	0.93	0.06	0.63	0.16	0.73	0.16
			50	5	0.91	0.07	0.54	0.18	0.56	0.18
	1 year	250	25	10	0.87	0.08	0.56	0.14	0.72	0.16
			50	5	0.84	0.09	0.47	0.16	0.52	0.17
RandomForest (S)	1 month	250	25	10	0.86	0.13	0.70	0.21	0.87	0.19
			50	5	0.91	0.11	0.60	0.33	0.72	0.34
	6 months	250	25	10	0.89	0.12	0.65	0.21	0.78	0.20
			50	5	0.94	0.09	0.75	0.24	0.84	0.22
	1 year	250	25	10	0.91	0.10	0.68	0.27	0.83	0.27
			50	5	0.91	0.09	0.69	0.19	0.85	0.19
SVC (S)	1 month	250	25	10	0.83	0.17	0.71	0.18	0.96	0.09
			50	5	0.95	0.07	0.71	0.31	0.77	0.32
	6 months	250	25	10	0.74	0.21	0.49	0.30	0.57	0.33
			50	5	0.98	0.03	0.85	0.18	0.92	0.15
	1 year	250	25	10	0.87	0.14	0.73	0.22	0.92	0.18
			50	5	0.93	0.08	0.64	0.27	0.74	0.29

Extra Feature Experiments

This chapter consists of result table scenarios with Extra Features: number of likes, followings, and followers.

First Synthetic Type

Table A.9 Model Results with First Scenario & Extra Features, Number of Synthetic Data:50
Unsupervised: U, Supervised: S

			ROC		NDCG@50		PRN@50	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	50	0.85	0.07	0.03	0.06	0.06	0.14
	6 months	50	0.79	0.10	0.03	0.03	0.07	0.12
	1 year	50	0.72	0.12	0.03	0.03	0.06	0.11
ECOD (U)	1 month	50	0.75	0.09	0.01	0.03	0.01	0.03
	6 months	50	0.59	0.13	0.00	0.01	0.00	0.01
	1 year	50	0.51	0.16	0.00	0.01	0.00	0.01
HDBSCAN (U)	1 month	50	0.64	0.10	0.21	0.17	0.29	0.25
	6 months	50	0.60	0.11	0.06	0.09	0.08	0.12
	1 year	50	0.57	0.12	0.04	0.06	0.06	0.12
IForest (U)	1 month	50	0.75	0.12	0.01	0.03	0.02	0.05
	6 months	50	0.64	0.14	0.00	0.02	0.01	0.03
	1 year	50	0.59	0.16	0.01	0.02	0.01	0.03
OCSVM (U)	1 month	50	0.86	0.08	0.03	0.04	0.07	0.10
	6 months	50	0.79	0.10	0.03	0.03	0.07	0.13
	1 year	50	0.72	0.12	0.03	0.03	0.07	0.12
RandomForest (S)	1 month	50	0.94	0.07	0.62	0.33	0.71	0.37
	6 months	50	0.97	0.03	0.45	0.39	0.48	0.42
	1 year	50	0.98	0.03	0.68	0.24	0.75	0.26
SVC (S)	1 month	50	0.97	0.06	0.69	0.33	0.76	0.35
	6 months	50	0.81	0.25	0.37	0.39	0.39	0.42
	1 year	50	0.96	0.06	0.54	0.31	0.57	0.34

Table A.10 Model Results with First Scenario & Extra Features, Number of Synthetic Data:250
Unsupervised: U, Supervised: S

			ROC		NDCG@250		PRN@250	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	250	0.78	0.15	0.08	0.07	0.07	0.06
	6 months	250	0.76	0.12	0.09	0.07	0.08	0.05
	1 year	250	0.70	0.12	0.09	0.07	0.08	0.06
ECOD (U)	1 month	250	0.68	0.14	0.02	0.04	0.02	0.03
	6 months	250	0.54	0.15	0.01	0.03	0.01	0.02
	1 year	250	0.45	0.16	0.01	0.01	0.01	0.01
HDBSCAN (U)	1 month	250	0.63	0.08	0.27	0.11	0.39	0.18
	6 months	250	0.61	0.08	0.17	0.11	0.19	0.14
	1 year	250	0.59	0.10	0.12	0.09	0.13	0.12
IForest (U)	1 month	250	0.67	0.17	0.04	0.04	0.04	0.05
	6 months	250	0.48	0.21	0.02	0.04	0.02	0.03
	1 year	250	0.45	0.21	0.02	0.03	0.02	0.03
OCSVM (U)	1 month	250	0.88	0.08	0.23	0.15	0.19	0.12
	6 months	250	0.79	0.11	0.12	0.10	0.11	0.08
	1 year	250	0.70	0.12	0.09	0.07	0.09	0.06
RandomForest (S)	1 month	250	0.98	0.06	0.89	0.18	0.94	0.16
	6 months	250	0.97	0.04	0.79	0.25	0.85	0.26
	1 year	250	0.92	0.11	0.58	0.36	0.60	0.38
SVC (S)	1 month	250	0.99	0.02	0.86	0.20	0.89	0.21
	6 months	250	0.97	0.09	0.78	0.33	0.81	0.33
	1 year	250	0.82	0.29	0.55	0.41	0.56	0.43

Table A.11 Model Results with First Scenario & Extra Features, Number of Synthetic Data:500
Unsupervised: U, Supervised: S

			ROC		NDCG@500		PRN@500	
CLF	V	SC	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	500	0.68	0.22	0.13	0.14	0.12	0.13
	6 months	500	0.68	0.18	0.12	0.09	0.09	0.07
	1 year	500	0.68	0.14	0.17	0.12	0.14	0.10
ECOD (U)	1 month	500	0.56	0.21	0.03	0.04	0.02	0.03
	6 months	500	0.44	0.19	0.01	0.02	0.01	0.02
	1 year	500	0.38	0.18	0.01	0.02	0.01	0.02
HDBSCAN (U)	1 month	500	0.60	0.09	0.28	0.10	0.39	0.16
	6 months	500	0.61	0.09	0.24	0.10	0.27	0.14
	1 year	500	0.58	0.09	0.19	0.10	0.19	0.12
IForest (U)	1 month	500	0.53	0.24	0.05	0.06	0.05	0.05
	6 months	500	0.35	0.22	0.02	0.03	0.02	0.03
	1 year	500	0.34	0.23	0.02	0.03	0.02	0.03
OCSVM (U)	1 month	500	0.87	0.08	0.40	0.17	0.35	0.15
	6 months	500	0.79	0.11	0.24	0.16	0.21	0.14
	1 year	500	0.69	0.14	0.17	0.12	0.15	0.10
RandomForest (S)	1 month	500	0.99	0.02	0.90	0.15	0.93	0.15
	6 months	500	0.97	0.08	0.79	0.30	0.81	0.30
	1 year	500	0.85	0.09	0.70	0.16	0.88	0.13
SVC (S)	1 month	500	0.98	0.04	0.86	0.19	0.89	0.20
	6 months	500	0.96	0.09	0.77	0.32	0.79	0.33
	1 year	500	0.96	0.11	0.80	0.27	0.81	0.28

Second Synthetic Type

Table A.12 Model Results with Second Scenario & Extra Features, Number of Synthetic Data:50
Unsupervised: U, Supervised: S

			ROC		NDCG@50		PRN@50	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	5	10	0.95	0.03	0.19	0.22	0.25	0.28
	10	5	0.92	0.04	0.08	0.11	0.15	0.24
ECOD (U)	5	10	0.99	0.01	0.52	0.15	0.58	0.17
	10	5	0.98	0.01	0.40	0.15	0.46	0.17
HDBSCAN (U)	5	10	0.50	0.12	0.22	0.17	0.38	0.30
	10	5	0.48	0.11	0.13	0.13	0.23	0.24
IForest (U)	5	10	0.92	0.04	0.02	0.06	0.02	0.05
	10	5	0.91	0.04	0.02	0.06	0.02	0.06
OCSVM (U)	5	10	0.95	0.02	0.06	0.08	0.10	0.14
	10	5	0.92	0.04	0.04	0.05	0.08	0.11
RandomForest (S)	5	10	0.99	0.04	0.91	0.13	0.94	0.11
	10	5	0.98	0.05	0.88	0.09	0.92	0.06
SVC (S)	5	10	0.98	0.07	0.87	0.21	0.91	0.20
	10	5	0.99	0.03	0.83	0.14	0.88	0.13

Table A.13 Model Results with Second Scenario & Extra Features, Number of Synthetic Data:250
Unsupervised: U, Supervised: S

			ROC		NDCG@250		PRN@250	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	25	10	0.89	0.09	0.25	0.22	0.23	0.23
	50	5	0.86	0.08	0.19	0.10	0.16	0.10
ECOD (U)	25	10	0.94	0.06	0.47	0.11	0.47	0.13
	50	5	0.92	0.06	0.39	0.10	0.40	0.11
HDBSCAN (U)	25	10	0.54	0.07	0.28	0.11	0.44	0.20
	50	5	0.54	0.07	0.22	0.11	0.31	0.17
IForest (U)	25	10	0.86	0.08	0.10	0.11	0.09	0.10
	50	5	0.85	0.07	0.10	0.10	0.09	0.09
OCSVM (U)	25	10	0.93	0.05	0.33	0.14	0.29	0.14
	50	5	0.88	0.07	0.23	0.11	0.20	0.10
RandomForest (S)	25	10	0.95	0.07	0.88	0.14	0.95	0.12
	50	5	0.93	0.13	0.70	0.36	0.76	0.36
SVC (S)	25	10	0.98	0.07	0.93	0.17	0.95	0.15
	50	5	0.97	0.07	0.84	0.24	0.90	0.21

Table A.14 Model Results with Second Scenario & Extra Features, Number of Synthetic Data:500
Unsupervised: U, Supervised: S

			ROC		NDCG@500		PRN@500	
CLF	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	50	10	0.82	0.15	0.35	0.23	0.31	0.24
	100	5	0.79	0.11	0.28	0.12	0.25	0.12
ECOD (U)	50	10	0.86	0.13	0.43	0.10	0.41	0.12
	100	5	0.85	0.12	0.38	0.08	0.37	0.09
HDBSCAN (U)	50	10	0.56	0.07	0.33	0.11	0.47	0.17
	100	5	0.56	0.07	0.27	0.11	0.35	0.16
IForest (U)	50	10	0.77	0.15	0.16	0.11	0.14	0.10
	100	5	0.77	0.13	0.15	0.10	0.13	0.09
OCSVM (U)	50	10	0.90	0.07	0.48	0.11	0.43	0.11
	100	5	0.82	0.10	0.34	0.10	0.31	0.10
RandomForest (S)	50	10	0.87	0.17	0.75	0.26	0.92	0.19
	100	5	0.93	0.12	0.81	0.19	0.91	0.16
SVC (S)	50	10	0.99	0.03	0.94	0.11	0.97	0.10
	100	5	0.98	0.04	0.84	0.22	0.87	0.21

Combined Synthetic Type

Table A.15 Model Results with Combined Scenario & Extra Features, Number of Synthetic Data:100
Unsupervised: U, Supervised: S

					ROC		NDCG@100		PRN@100	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	50	5	10	0.90	0.05	0.16	0.13	0.24	0.23
			10	5	0.88	0.05	0.09	0.07	0.14	0.17
	6 months	50	5	10	0.87	0.06	0.17	0.13	0.28	0.25
			10	5	0.86	0.06	0.10	0.08	0.17	0.19
	1 year	50	5	10	0.84	0.07	0.18	0.14	0.29	0.25
			10	5	0.83	0.07	0.09	0.08	0.16	0.20
ECOD (U)	1 month	50	5	10	0.85	0.05	0.41	0.10	0.54	0.15
			10	5	0.86	0.05	0.34	0.10	0.42	0.14
	6 months	50	5	10	0.77	0.07	0.42	0.10	0.56	0.15
			10	5	0.78	0.07	0.35	0.10	0.43	0.15
	1 year	50	5	10	0.74	0.09	0.42	0.10	0.56	0.15
			10	5	0.74	0.08	0.35	0.11	0.43	0.15
HDBSCAN (U)	1 month	50	5	10	0.58	0.08	0.30	0.13	0.49	0.22
			10	5	0.56	0.08	0.25	0.14	0.37	0.23
	6 months	50	5	10	0.56	0.08	0.22	0.12	0.40	0.25
			10	5	0.54	0.08	0.16	0.12	0.25	0.21
	1 year	50	5	10	0.55	0.09	0.20	0.12	0.39	0.25
			10	5	0.53	0.08	0.14	0.11	0.25	0.21
IForest (U)	1 month	50	5	10	0.83	0.06	0.05	0.08	0.05	0.07
			10	5	0.82	0.06	0.05	0.08	0.05	0.08
	6 months	50	5	10	0.78	0.07	0.05	0.09	0.05	0.08
			10	5	0.77	0.07	0.05	0.08	0.05	0.08
	1 year	50	5	10	0.76	0.08	0.06	0.09	0.06	0.09
			10	5	0.76	0.08	0.05	0.08	0.05	0.07
OCSVM (U)	1 month	50	5	10	0.90	0.04	0.11	0.09	0.13	0.10
			10	5	0.89	0.04	0.08	0.07	0.11	0.08
	6 months	50	5	10	0.87	0.05	0.10	0.09	0.13	0.11
			10	5	0.86	0.06	0.08	0.06	0.11	0.09
	1 year	50	5	10	0.84	0.06	0.11	0.08	0.13	0.10
			10	5	0.83	0.07	0.08	0.06	0.10	0.09
RandomForest (S)	1 month	50	5	10	0.97	0.05	0.70	0.23	0.83	0.23
			10	5	0.96	0.05	0.73	0.16	0.87	0.14
	6 months	50	5	10	0.93	0.04	0.56	0.16	0.84	0.24
			10	5	0.97	0.04	0.64	0.24	0.76	0.24
	1 year	50	5	10	0.95	0.06	0.70	0.14	0.87	0.14
			10	5	0.96	0.04	0.48	0.33	0.59	0.36
SVC (S)	1 month	50	5	10	0.88	0.16	0.64	0.28	0.79	0.31
			10	5	0.97	0.04	0.76	0.19	0.88	0.15
	6 months	50	5	10	0.73	0.15	0.53	0.23	0.79	0.31
			10	5	0.93	0.07	0.54	0.25	0.70	0.31
	1 year	50	5	10	0.96	0.04	0.62	0.23	0.77	0.25
			10	5	0.95	0.04	0.53	0.27	0.65	0.32

Table A.16 Model Results with Combined Scenario & Extra Features, Number of Synthetic Data:500
Unsupervised: U, Supervised: S

					ROC		NDCG@500		PRN@500	
CLF	V	SC	N	M	Mean	SD	Mean	SD	Mean	SD
DBSCAN (U)	1 month	250	25	10	0.76	0.17	0.23	0.11	0.19	0.10
			50	5	0.79	0.12	0.25	0.09	0.21	0.09
	6 months	250	25	10	0.80	0.11	0.32	0.12	0.30	0.16
			50	5	0.80	0.10	0.28	0.10	0.25	0.09
	1 year	250	25	10	0.79	0.09	0.35	0.13	0.34	0.19
			50	5	0.78	0.09	0.29	0.10	0.26	0.10
ECOD (U)	1 month	250	25	10	0.76	0.12	0.39	0.07	0.44	0.11
			50	5	0.78	0.11	0.34	0.07	0.36	0.09
	6 months	250	25	10	0.70	0.11	0.41	0.07	0.47	0.12
			50	5	0.71	0.11	0.36	0.07	0.39	0.10
	1 year	250	25	10	0.67	0.11	0.41	0.07	0.48	0.12
			50	5	0.68	0.11	0.36	0.07	0.40	0.10
HDBSCAN (U)	1 month	250	25	10	0.60	0.06	0.38	0.10	0.54	0.15
			50	5	0.60	0.06	0.35	0.10	0.49	0.16
	6 months	250	25	10	0.59	0.06	0.34	0.10	0.46	0.16
			50	5	0.59	0.06	0.31	0.11	0.39	0.16
	1 year	250	25	10	0.59	0.07	0.32	0.11	0.44	0.16
			50	5	0.58	0.06	0.29	0.12	0.36	0.16
IForest (U)	1 month	250	25	10	0.73	0.14	0.15	0.10	0.13	0.09
			50	5	0.74	0.12	0.15	0.11	0.13	0.10
	6 months	250	25	10	0.66	0.14	0.16	0.12	0.14	0.11
			50	5	0.65	0.13	0.16	0.11	0.13	0.10
	1 year	250	25	10	0.66	0.13	0.17	0.12	0.15	0.11
			50	5	0.65	0.13	0.17	0.12	0.15	0.11
OCSVM (U)	1 month	250	25	10	0.89	0.06	0.45	0.13	0.41	0.13
			50	5	0.87	0.07	0.39	0.13	0.35	0.13
	6 months	250	25	10	0.85	0.08	0.41	0.12	0.38	0.12
			50	5	0.83	0.08	0.34	0.12	0.30	0.11
	1 year	250	25	10	0.82	0.08	0.39	0.11	0.38	0.12
			50	5	0.79	0.09	0.32	0.11	0.29	0.11
RandomForest (S)	1 month	250	25	10	0.97	0.05	0.81	0.24	0.86	0.25
			50	5	0.95	0.06	0.78	0.24	0.87	0.24
	6 months	250	25	10	0.93	0.08	0.71	0.18	0.84	0.17
			50	5	0.95	0.07	0.74	0.18	0.85	0.15
	1 year	250	25	10	0.88	0.09	0.69	0.18	0.84	0.18
			50	5	0.93	0.06	0.73	0.17	0.88	0.15
SVC (S)	1 month	250	25	10	0.96	0.08	0.79	0.28	0.84	0.29
			50	5	0.98	0.04	0.87	0.19	0.93	0.18
	6 months	250	25	10	0.91	0.09	0.70	0.23	0.90	0.20
			50	5	0.95	0.07	0.77	0.20	0.88	0.17
	1 year	250	25	10	0.83	0.17	0.69	0.18	0.87	0.16
			50	5	0.92	0.09	0.70	0.25	0.79	0.26

Real World Use Cases

Example Figures for politician's Anomaly & Bot scores and their correlations are shown below figures.

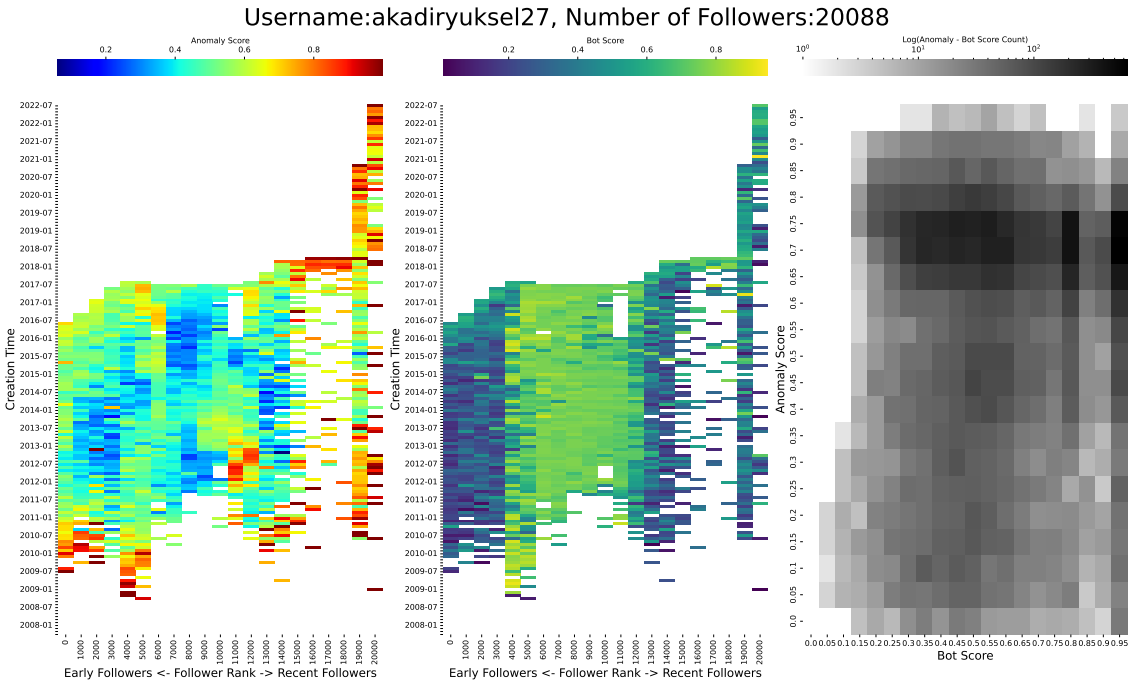

Figure A.5 Binned Creation Time - Rank Heatmap Figure

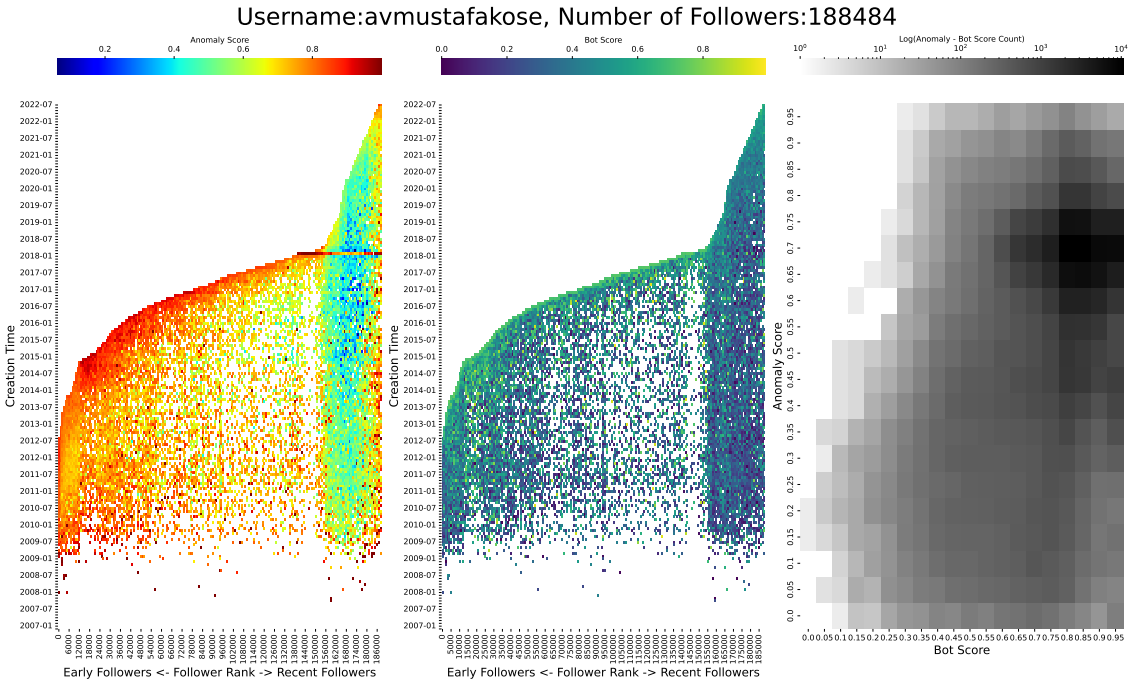

Figure A.6 Binned Creation Time - Rank Heatmap Figure

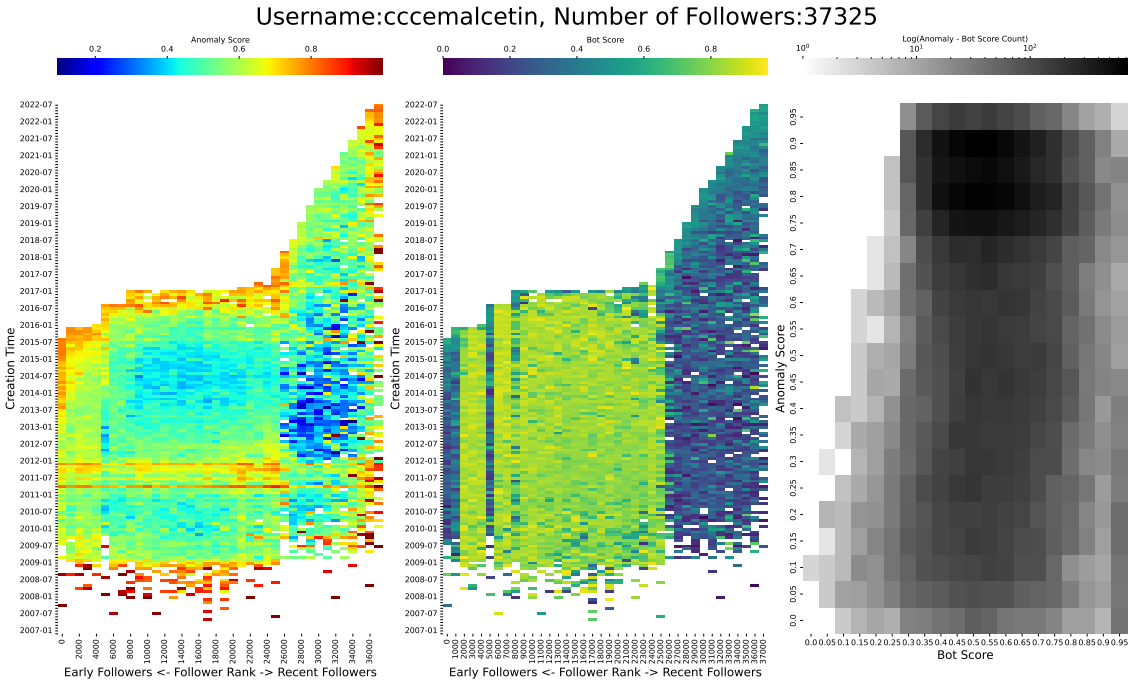

Figure A.7 Binned Creation Time - Rank Heatmap Figure

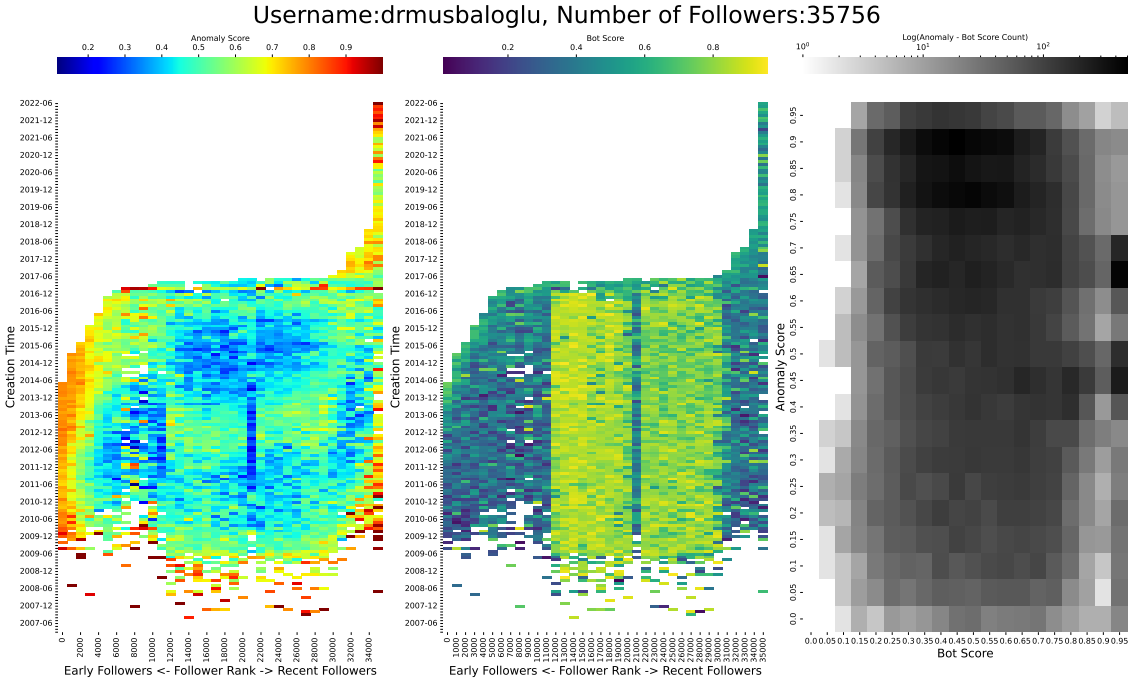

Figure A.8 Binned Creation Time - Rank Heatmap Figure

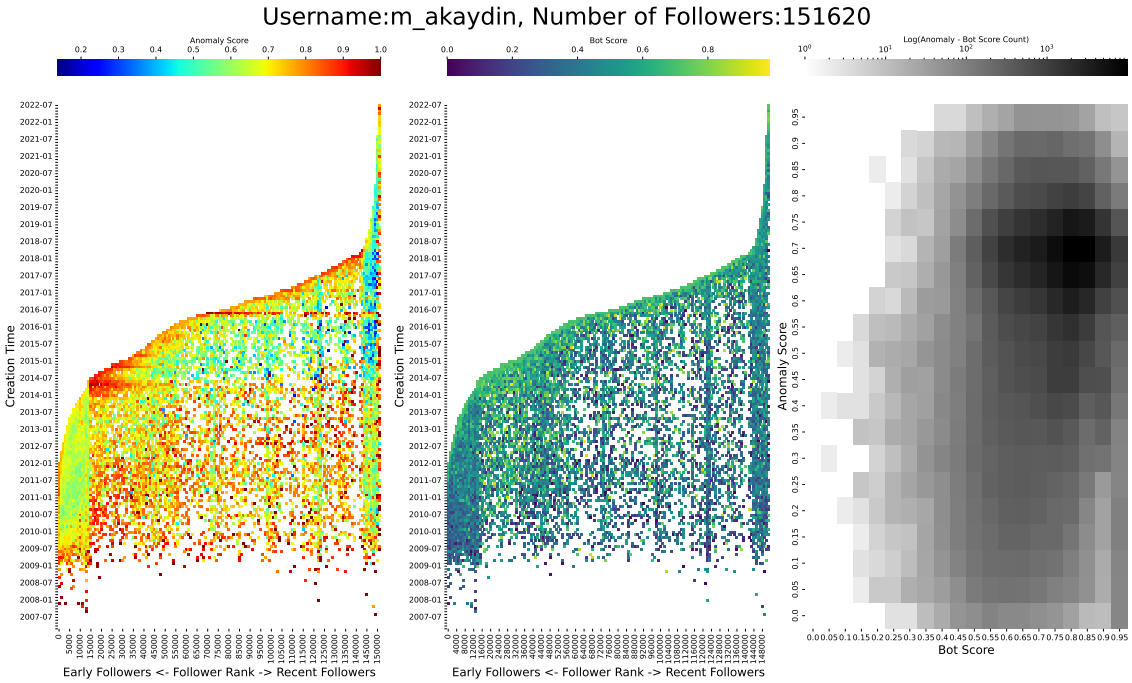

Figure A.9 Binned Creation Time - Rank Heatmap Figure

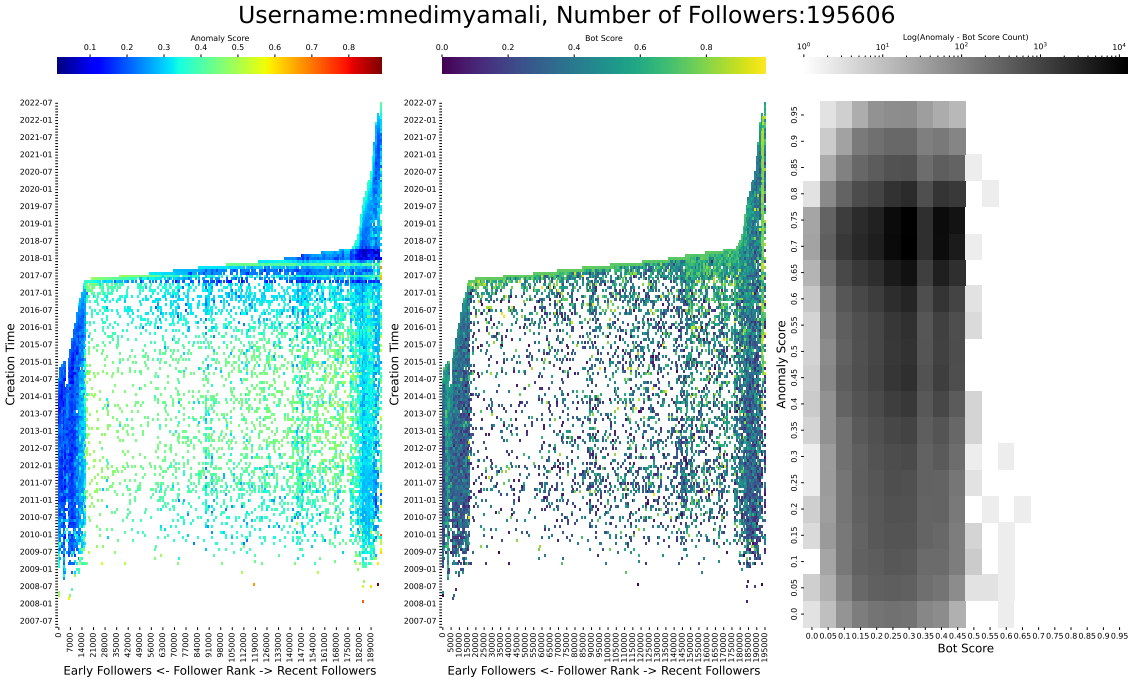

Figure A.10 Binned Creation Time - Rank Heatmap Figure

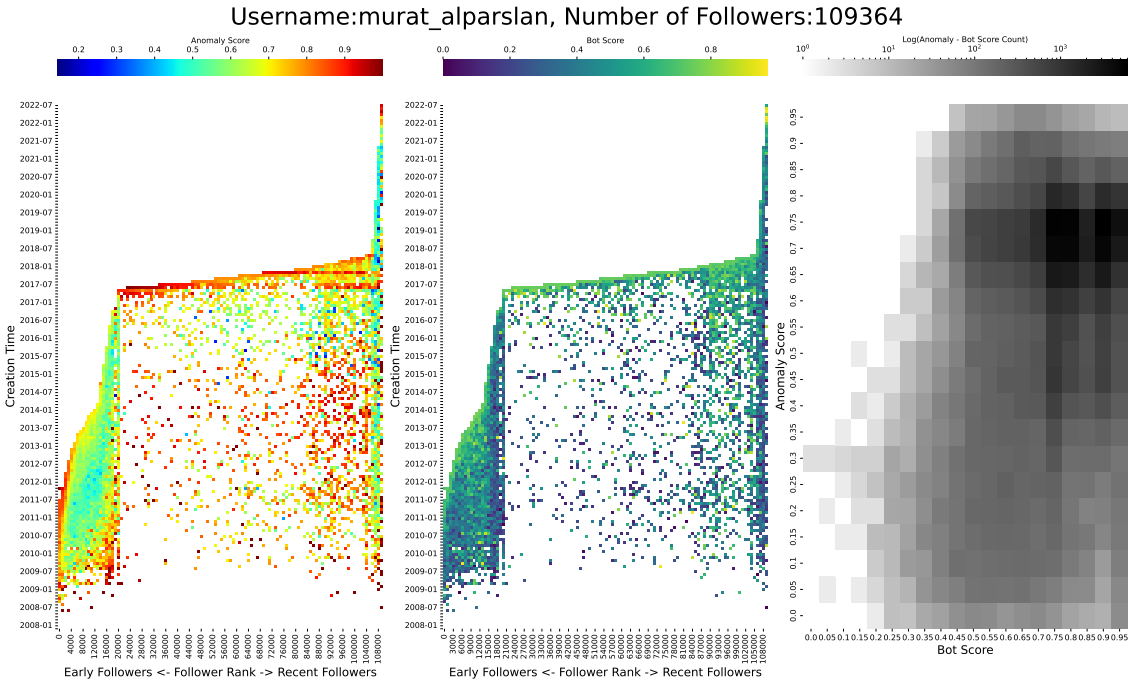

Figure A.11 Binned Creation Time - Rank Heatmap Figure

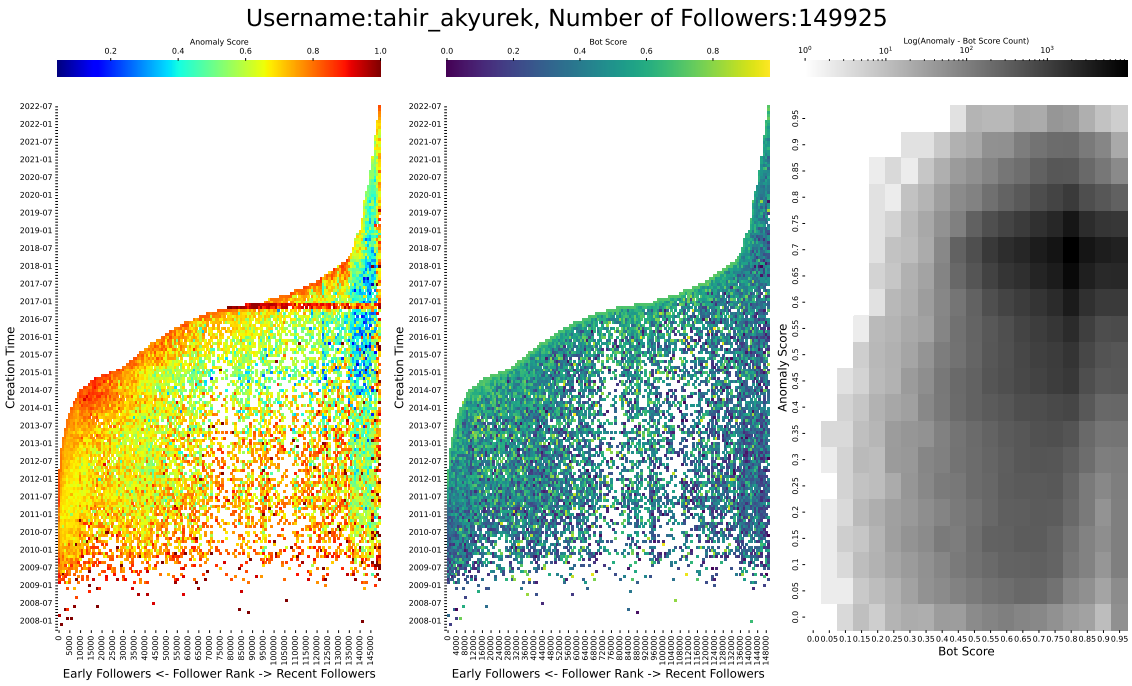

Figure A.12 Binned Creation Time - Rank Heatmap Figure

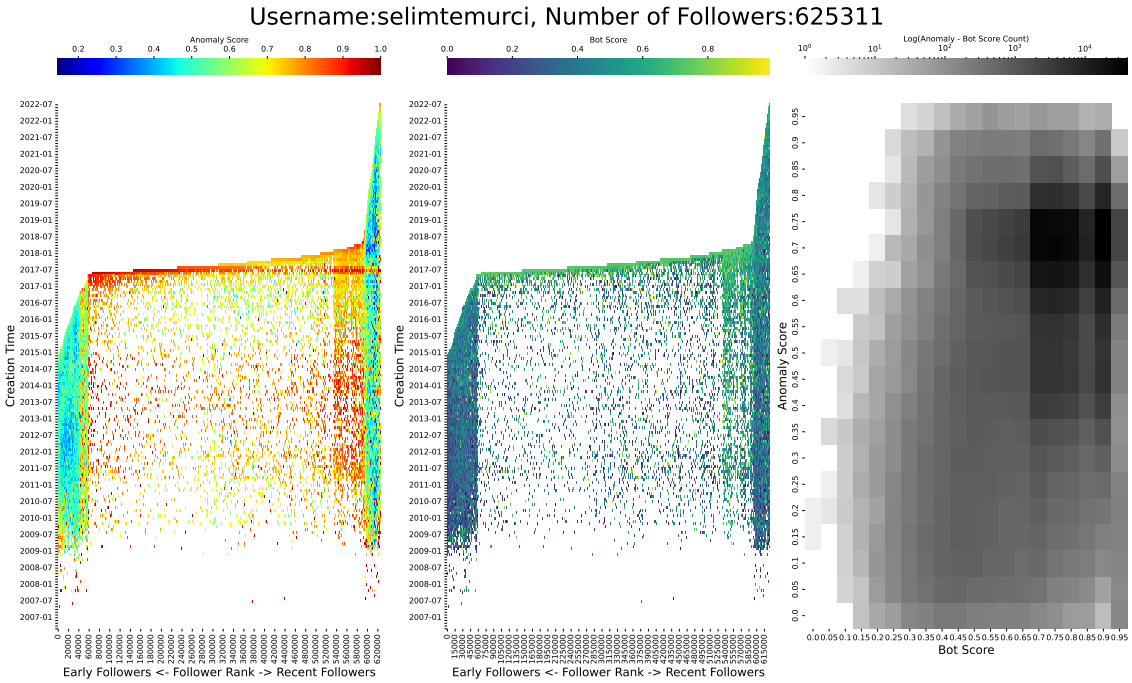

Figure A.13 Binned Creation Time - Rank Heatmap Figure

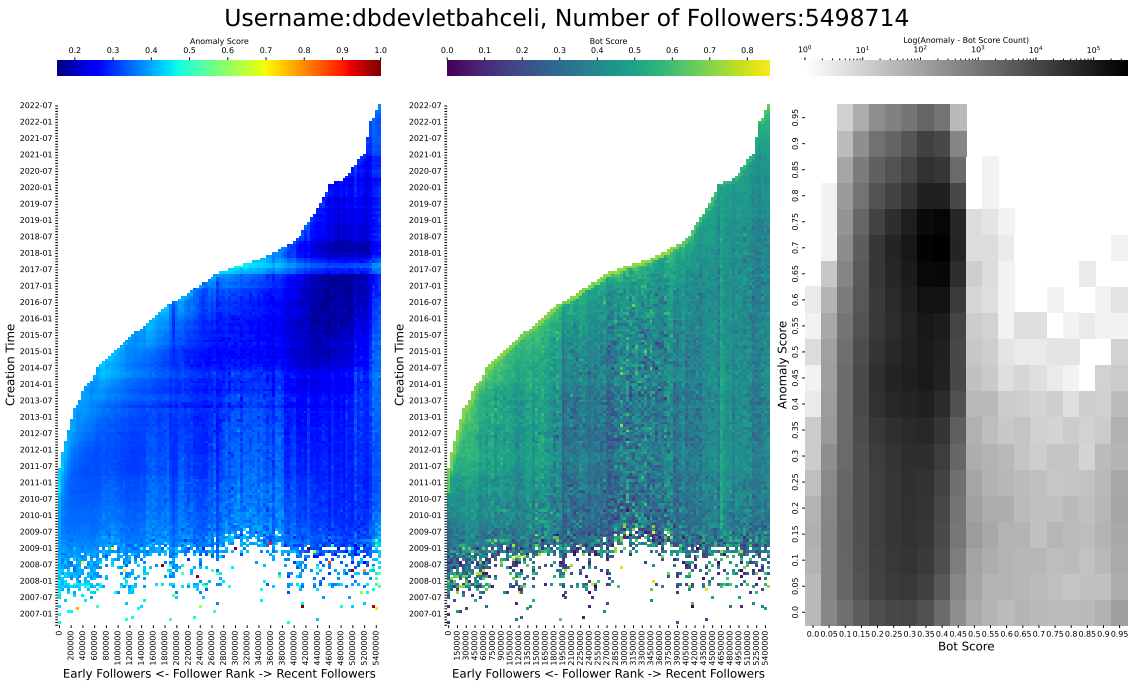

Figure A.14 Binned Creation Time - Rank Heatmap Figure

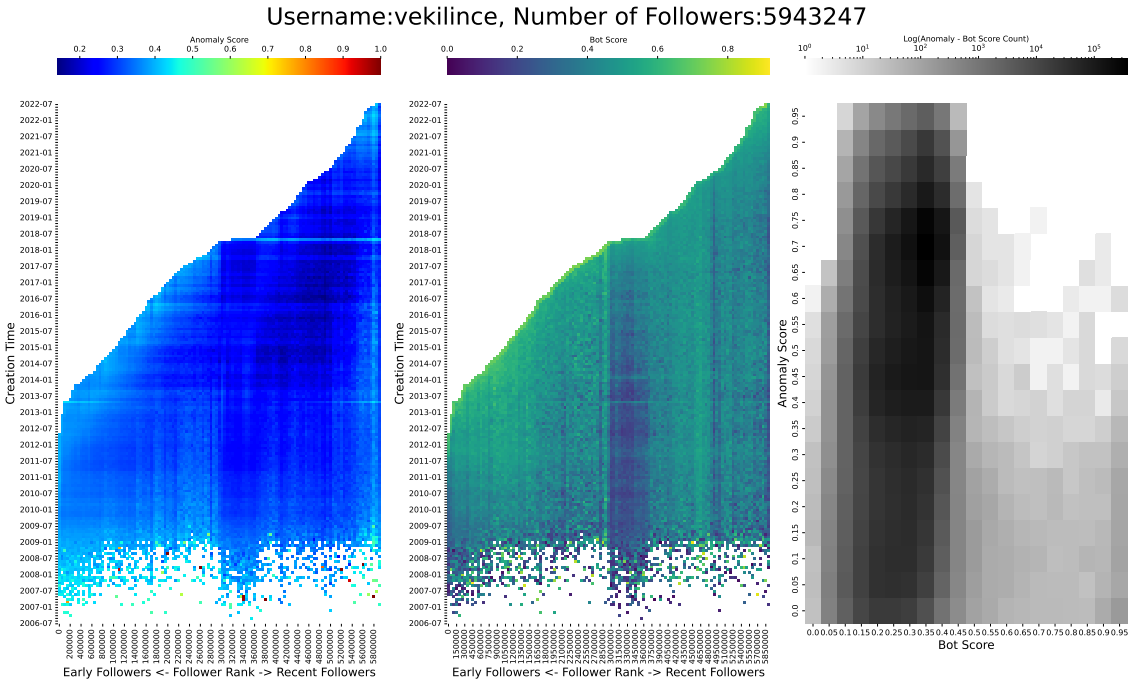

Figure A.15 Binned Creation Time - Rank Heatmap Figure