Graph embeddings analysis in Social Networks

Ioannis Kontogiorgakis

Thesis submitted in partial fulfillment of the requirements for the

Bachelor's degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor:
Prof. *Sotiris Ioannidis*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

Graph embeddings analysis in Social Networks

Thesis submitted by
Giannis Kontogiorgakis
in partial fulfillment of the requirements for the
Bachelor's degree in Computer Science

THESIS APPROVAL

Author: ______________________________
Giannis Kontogiorgakis

Committee approvals: ______________________________
Polyvios Pratikakis
Assistant Professor, Thesis Supervisor, Committee Member

____
Sotiris Ioannidis
Associated Professor, Thesis Advisor, Committee Member

____
Ioannis Tzitzikas
Professor, Committee Member

Departmental approval: ______________________________
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, September 2022

Graph embeddings analysis in Social Networks

Abstract

Nowadays, social media are becoming more and more part of our lives, as users grow exponentially on every platform. Twitter, being one of the most popular social media platforms, allows users to debate on various topics and express their opinion. However, offensive behaviour, hate and controversial comments, propaganda or any other violation of Twitter rules surrounding abuse, often leads to user account suspension or permanent account deletion.

On February 2022 Russia invaded Ukraine, causing the well-known Russo-Ukraine war. This act directly triggered turmoil on social media platforms, since users all over the world debate and offer their opinion on this matter. Posts on Twitter, are often accompanied by hashtags relevant to the content of the post. Throughout this period, from February to June, in which war was at its climax, we collected data , originated from 8 million users. Exploiting Twitter open API, we extracted the social relations between users in graph representations.

In this thesis, we seek to answer the question "Is it possible to predict a Twitter user account suspension, based solely on social relations of the user ?". We present a machine learning pipeline, in which we analyze how well a machine learning model can predict user suspension, and how the accuracy of the model changes during the period February to June.

Ανάλυση ενσωματωμένων γράφων στα μέσα κοινωνικής δικτύωσης

Περίληψη

Στις μέρες μας, τα μέσα κοινωνικής δικτύωσης γίνονται ολοένα και περισσότερο μέρος της καθημερινότητας μας, καθώς οι χρήστες αυξάνονται εκθετικά σε κάθε πλατφόρμα. Το Twitter, όντας μια από τις πιο δημοφιλείς πλατφόρμες κοινωνικής δικτύωσης, δίνει την ευκαιρία στους χρήστες της να συζητούν ποικίλα θέματα και να εκφέρουν τις απόψεις τους. Ωστόσο,η προσβλητική συμπεριφορά, το μίσος, αμφιλεγόμενα σχόλια, η προπαγάνδα καθώς και οποιαδήποτε άλλη παράβαση της πολιτικής του Twitter όσον αφορά την ύβρη, συχνά οδηγεί σε παύση ή και σε διαγραφή του λογαριασμού του χρήστη.

Στις 22 Φεβρουαρίου 2022, η Ρωσία εισέβαλε στην Ουκρανία προκαλώντας τον, σε όλους γνωστό πλέον, Ρωσσο-Ουκρανικό πόλεμο. Αυτή πράξη, προκάλεσε αναταραχή στα μέσα κοινωνικής δικτύωσης, καθώς χρήστες από όλο τον κόσμο έσπευσαν να συζητήσουν και να προβάλλουν την άποψη τους πάνω σε αυτό το θέμα. Οι δημοσιεύσεις στο Twitter συχνά συνοδεύονται από hashtags σχετικά με το περιεχόμενο της δημοσίευσης. Κατά την διάρκεια της περιόδου, Φεβρυοάριο έως Ιούνιο,στην οποία ο πόλεμος βρισκόταν στο αποκορύφωμα του, συλλέξαμε δεδομένα από 8 εκατομμύρια χρήστες. Εκμεταλλευόμενοι το ανοιχτό API του Twitter, εξαγάγαμε τις κοινωνικές σχέσεις μεταξύ των χρηστών σε μορφή γράφου.

Σε αυτήν την έρευνα, ψάχνουμε απάντηση στο ερώτημα: Είναι δυνατό να προβλέψουμε την παύση ενός Twitter λογιαριασμού χρήστη, χρησιμοποιώντας μόνο τις κοινωνικές σχέσεις του χρήστη· Σας παρουσιάζουμε ένα pipeline μηχανικής μάθησης, στο οποίο αναλύουμε κατά πόσο ένα μοντέλο μηχανικής μάθησης μπορεί να προβλέψει την παύση του λογαριασμού ενός χρήστη, και πως αλλάζει η ακρίβεια του μοντέλου κατά την διάρκεια της περιόδου Φεβρουάριο έως Απρίλιο.

στους γονείς μου

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Social media growth brought major changes in many aspects of our every day lives. Although their emergence initially served communication purposes between registered users, social media now evolved to be one of the most widespread means in terms of information. Major news topics, such as the COVID pandemic, Russo-Ukrainian war, Presidential elections and significant political matters, attract the interest of many users around the world. Different users can now express their opinions on the matter, comment on other user's opinion and agree or disagree with their points of view. Although, the controversial nature of these topics, often results in misconduct and offensive behaviour. Currently, big social media platforms like Facebook, Twitter tend to suspend accounts that violate their rules concerning user malicious behaviour.

Social media platforms allow registered users to interact with each other. These interactions can be construed in terms of social relations. Within these bounds, a user can add/follow another user but also can retweet, mention, and quote a post originating from other users. Currently, many user account suspension prediction methods have been designed using Machine Learning. Although, these methods are designed to apply to a specified language and are based on the user's textual features, such as created posts.

In my thesis, we attempt to predict a Twitter user account suspension based on the the social relations of the user(quote, retweet, mention). For this purpose, we implement various scenarios, utilizing multiple datasets of active, suspended and deleted users in order to produce a valid representation of user suspension prediction.

Although graph representation is clear and easily understandable, its utility in machine learning is challenging, as mathematical and statistical operations are limited. A solution to this problem, appears to be graph embeddings. Each node is encoded to a numerical vector representation, maintaining graph topology and vertex-to-vertex relationship of the initial graph form. Furthermore, vector spaces have a richer toolset of approaches in terms of machine learning techniques, in contrast to graph representation.

In this thesis, we manage to examine the possibility of malicious account detection based on evolving Twitter social graphs. We manage to develop and test multiple machine learning models and approaches in order to identify the best solution. In the following sections, we discuss and analyze the implementation and performance of different scenarios, and how the accuracy of our model varies throughout a long time period.

Chapter 2

Related Work

Similar work on Twitter user account suspension focuses on analysis of the factors that lead to account suspension and deletion in regard to Twitter policy rules, as cited in [5], [26], [21], [6] . One of the most fundamental studies in terms of suspended account analysis is Vern Paxson's "Suspended accounts in retrospect: an analysis of twitter spam" [24]. For the purpose of the analysis, they identified 1.1 million suspended accounts and extracted a collection of 1.8 billion tweets in seven months period, 80 million of which belong to spam accounts. A similar research [25] conducted, based on BlackLivesMatter movement, examining the factors for undesirable behavior in terms of spamming, negative language, hate speech, and misinformation spread. This study shown that users who participated in 2020 BlackLivesMatter discussion have more negative and undesirable tweets, compared to the users who did not. Some researches, attempt to detect propaganda and mudslinging during presidential elections [10], [4], fake news [20], [19], or even spam campaigns [2], [27]. In this context, many studies managed to analyze and label abusive behaviour on Twitter social platform [12], [7], [18]. Furthermore, a plethora of studies [13], [17], [11], focus on sentiment analysis on tweets, originating from online social users. For example, in [17], a large set of manually labeled tweets in different languages is analyzed, in order to classify users in sentiment classes (negative, neutral and positive)

A great number of studies, utilize Machine Learning and Neural Networks implementation in order to identify automatically managed accounts, also known as bots, [9], [22], [3], [8]. In a similar topic to our study, [23] , they utilize graph embeddings in order to detect bots on Twitter social platform. The contribution of this study, is the creation of a stacking-based ensemble in order to exploit text and graph features. A similar approach is also used in Ilia Karpov's "Detecting automatically managed accounts in online social networks" [14], where traditional graph embeddings techniques, such as RandomWalk, Node2Vec and Attri2Vec are utilized for classification between human and artificial accounts. In addition, in another research, [16], a multi-regional dataset of tweets originating from rebels, normal and counter rebel users from five countries is utilized for the purpose of user classification based on their political front, using graph embeddings as input to machine learning algorithms.

Chapter 3

Data

For the purpose of my thesis, we collected public user information, concerning RussoUkraine war. Due to the huge impact of this matter in many aspects of human life, with regard to the controversy between different sides, we observed a barrage of user interaction on Twitter social platform. Utilizing Twitter API, we extracted 57 million tweets, originated from 8.351.592 million users throughout the period 23 February 2022 to 23 June 2022, pursuant to input query parameters. Concerning query parameters, we used the most trending, popular and relevant hashtags, corresponding to our topic. In Table 3.1, we present the set of hashtags we used for our data collection retrieval.

In order to identify suspended user accounts, we availed ourselves of one Twitter API features [1] , concerning the detection of the exact suspension day of the account. The retrieved collection consists of suspended, deactivated, deleted and private user accounts. For the purpose of labeling, we classified user status into two categories.

- suspended: all public user accounts that are marked as suspended. For the suspended users we use label 1.

- normal: all users's accounts that are not marked as suspended, deactivated, deleted or private. For normal users we use label 0.

Using this method, we detected 243.687 suspended and 8.107.905 normal user accounts throughout this period.

#Ukraine, #Ukraina, #ukraina, #Украина, #Украине, #PrayForUkraine, #UkraineRussie, #StandWithUkraine, #StandWithUkraineNOW, #RussiaUkraineConflict, #RussiaUkraineCrisis, #RussiaInvadedUkraine, #WWIII, #worldwar3, #Война, #BlockPutinWallets, #UkraineRussiaWar, #Putin, #Russia, #Россия, #StopPutin, #StopRussianAggression, #StopRussia, #Ukraine_Russia, #Russian_Ukrainian, #SWIFT, #NATO, #FuckPutin, #solidarityWithUkraine, #PutinWarCriminal, #PutinHitler, #BoycottRussia, #with_russia, #FUCK_NATO, #ЯпротивВойны, #StopNazism #myfriendPutin #UnitedAgainstUkraine #StopWar #ВпередРоссия, #ЯМыРоссия, #ВеликаяРоссия, #Путинмойпрезидент, #россиявперед, #россиявперёд, #ПутинНашПрезидент, #ЗаПутина, #Путинмойпрезидент, #ПутинВведиВойска, #СЛАВАРОССИИ, #СЛАВАВДВ

Table 3.1: Set of hashtags used in data collection query.

Chapter 4

Methodology

Our goal is to discover whether can we predict the account suspension of a Twitter user, based solely on his social relations(retweets, quotes and mentions) on Twitter social platform.

Our methodology focuses on a machine learning classification category using 4 popular models: Naive Bayes, Linear Regression, Random Forest and XGBoost. Besides model selection and initial evaluation of the best model, we are interested to identify how the user suspension prediction is affected by the time period and by graph knowledge growth. For this purpose, we extract 4 datasets, based on different time periods. Thus the datasets we extracted for training and evaluation of our model are:

- February - March: social relations of users from 23 of February 2022 to 23 of March 2022.
- February - April: social relations of users from 23 of February 2022 to 23 of April 2022.
- February - May: social relations of users from 23 of February 2022 to 23 of May 2022.
- February - June: social relations of users from 23 of February 2022 to 23 of June 2022.

Period	# of edges	# of nodes
February - March	63.981.724	7.295.679
February - April	77.935.623	8.042.713
February - May	85.445.261	8.424.198
February - June	89.743.715	8.648.253

Table 4.1: Total edges and nodes number of each graph

We managed to extract social relations of users from the social platform in the form of a triplet "x r y", where x represents the source user, namely the user that start the interaction, r represents the relation-interaction the user x triggers, and y serve for the destination user. We combine all relation, and store them in graph form. Since graphs need to have an appropriate format in order to fit our model, we utilize a machine learning Neural Network approach, in order to transform graphs into graph embeddings. As stated before, graph embeddings are numerical vector representations for each entity that exists in the graph. Utilizing Twitter API, we managed to identify suspended user accounts for each graph. We then combined the embeddings of each user into a tsv file, and appended a target integer for each user. Target value 1 means that the user account is suspended at the specified time period, while target value 0 means that the the account belongs to a normal user. We use tsv embeddings file as input for our model both for training and performance testing. For the purpose of my thesis, we examine 3 different scenarios in order to identify the best possible problem solution.

For our initial scenario, we want to discover how well the model trained in the initial data, can adapt on graph changes during different time periods. For that reason, we utilize the first graph (February - March) as train dataset for our model. We then measure the performance on each different time period while graph grows; February-March, February-April, February-May, February-June. We examine this particular scenario with the knowledge that, while graph evolve and new relations are generated, the outcome of graph embeddings may change and provide inconsistency between training and evaluation data portions.

Based on the knowledge of graph evolution, in the second scenario, we are interested to measure the performance with a more traditional machine learning approach where the model is re-trained for each different dataset. For this purpose, we split each dataset at rate 80/20 (train/test) and measure the performance over the test data portion. This particular approach allows the isolation of the noise that can be generated via graph evolution.

Finally, we manage to develop a scenario that can probably resolve the problems of the previous two scenarios. Initially, we train and evaluate our model on the initial first-month dataset and keep the users that were utilized in model training. Those users will contain our model training set during all datasets, with the only difference that model is re-trained over each different time period. With particular implementation we keep our model stable, since the training dataset is remaining the same, in the perspective of the users set, but combined with updated user embeddings it can overcome the issue of graph evolution.

Chapter 5

Implementation

As we mentioned earlier, our implementation consists of multiple steps that are required in order to solve the above presented-scenarios. Initially, we manage to extract user social relations in order to translate graph relations into vector form of scalar values. As a next step, we generate a machine learning pipeline with data pre-processing, feature selection, K-fold cross validation and model hyper-parameter fine-tuning which leads to our initial model selection. In this section, we discuss in detail each step of our implemented method.

5.1 Graph Embeddings

As stated before, graphs, cannot be used by a machine learning model an their initial form, due to its limited mathematical and statistical operations. The graphs we utilize for this thesis are multi-layer graphs, meaning that they contain more than one relation. In our case, the graph nodes (users) are connected with 3 different relations: retweet, quote and mention. Thus, our initial step is to transform Twitter multi-layer social graph into graph embeddings.

For this task, we tested various "traditional" graph embedding techniques for multi and single layer graphs. In particular, we tested DeepWalk, LINE, Node2Vec, SDNE and Struc2Vec. Although, due to their poor accuracy performance and time-consuming training in a deducted example dataset, we chose to solve this issue by using a Neural Network implementation, in which the model learns the relations between the nodes and creates a numerical vector representation for each user-node. These vectors will be then used as input in our machine learning model.

The perfect tool for this operation, proved to be Facebook's framework: PyTorch BigGraph [15]. Using graph partitioning, the model does not fully load in memory, allowing us to process graphs with million edges, such as our Twitter social graph. In addition,this implementation allows us to process both multi and single layer graphs as there is no limitation concerning the number of graph relations.

The next step was to decide the appropriate number of dimensions that each

Social Graph	Output dimension	MRR	AUC
Quote	50	0.65	0.81
Mention	50	0.54	0.87
Retweet	50	0.54	0.84
Multi-layer	50	0.79	0.96
Quote	100	0.67	0.78
Mention	100	0.53	0.87
Retweet	100	0.52	0.84
Multi-layer	100	0.82	**0.97**
Quote	150	0.66	0.78
Mention	150	0.52	0.86
Retweet	150	0.52	0.84
Multi-layer	150	**0.84**	**0.97**
Quote	200	0.66	0.77
Mention	200	0.52	0.85
Retweet	200	0.52	0.84
Multi-layer	200	0.54	0.80

Table 5.1: Graph embedding evaluation performance over different social relation graphs combined with different output dimensions. The highest performance is presented in bold text.

user's embedding vector should have. After testing several embeddings' output dimensions, we concluded that the output vector dimension equal to 150 is the best for our model, as MRR and AUC scores are higher (Table 5.1).

5.2 Data Pre-processing

Due to the enormous number of normal user accounts, model training would be uneven and extremely time- consuming. For that reason, dataset balance is deemed necessary. We used a random under-sampling technique to delete random normal user accounts, so that suspended users number is equal to normal users.

In order to avoid model overfitting, we apply feature selection. For this purpose, we used Lasso regression, to select the most significant features. Initially, we tune hyper-parameter alpha of the Lasso regression model. For this purpose, we utilize first-month graph embedding dataset (February-March). We split our dataset into 10 folds, using K-fold cross validation and we compute the mean squared error for each alpha parameter. The best alpha parameter is selected based on the the minimization of mean squared error, across all folds. After fine-tuning, we train Lasso Regression model on the best alpha parameter and we compute the coefficient of each feature of the dataset. Zero coefficient features, imply their minimum significance in the user's account suspension, and thus should be removed from the dataset. We later utilize those features, for our classification model hyper-parameter tuning , training and performance testing.

In order to make it easy for the model to understand our user suspension classification problem, we normalize our data before training. For this reason, we utilize MinMax scaler. Thus, the embeddings of each user are distributed in a range of 0 to 1. This step, ensures that our model avoids high generalization errors during training.

5.3 Model Selection

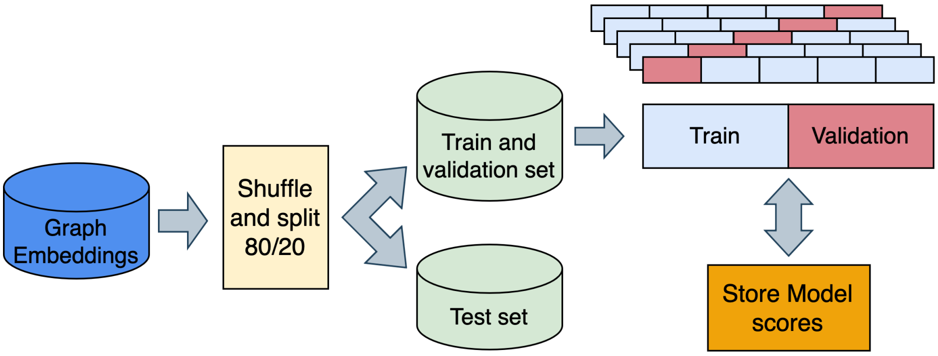

Figure 5.1: ML model selection pipeline

As shown in Figure 5.1, we try to implement a classification model, in order to predict user account suspension accuracy, based on the social relationships of the user. For this reason, we tested 4 different models, in order to select the best model for our classification problem. The classifiers we tested are XGBoost, Random Forest, Linear Regression and Naive Bayes. For the purpose of model selection, we utilize first-month graph dataset (February - March).

At first, we split our input dataset in train and validation sets, in a ratio of 80/20. The train set is used for model selection and hyper-parameters tuning, while the validation set is used for performance measurement. Implementing K-fold cross validation, we split our train set into 5 folds. We then, select the most significant features from feature selection. We train each classification model on a train set for each fold. In order to assure adjustment to the entire train set, we measure each model performance, by the average validation score across each fold. This score gives us a first impression of how the model adapts on the first-month dataset. For testing purposes, we also store the training f1 score, which is calculated by predicting the entire training set.

During the model selection step, we also tune the hyper-parameters of the models, in order to find the best parameters for training in the next step. For each model, and for each parameter set, we keep track of the f1 scores attained during hyper-parameters tuning. After this procedure, we attain the best model and its best parameters, which will be used for our final model training in the next steps. Statistics of model selection training are shown below (Table 5.2).

As it seems, from Table 5.2, the best model according to validation score is XGBoost. We, then, train our final model with the best parameters extracted from model fine-tuning, utilizing as train set the entire first-month dataset (February - March).

Model	Validation Score	Training Score
Naive Bayes	0.631	0.631
Linear Regression	0.672	0.67
Random Forest	0.723	0.97
XGB	**0.728**	0.99

Table 5.2: Train statistics of model selection and hyper-parameter fine-tuning. The highest performance is presented in bold text.

Chapter 6

Performance

In order to measure the performance of our model, we implemented 3 scenarios. In these scenarios, we utilize different train and test datasets, so that we analyze how the model adapts to different situations. For the purpose of the scenarios, we measured the exact number of users that have social interactions each month separately. Our measurements are the following:

Period	Normal Users	Suspended users	Total users
23 February - 23 March	6.868.232	165.627	7.033.859
24 March - 23 April	690.956	37.127	728.083
24 April - 23 May	340.761	29.923	370.684
24 May - 23 June	207.956	11.010	218.966

Table 6.1: Normal, suspended and total users count per month

In the following sections, we discuss in detail the steps and the results of our scenarios.

6.1 First scenario

For the first scenario, we want to test how well the model adapts at the first-month data. For the purpose of training our mode, we use the entire dataset of first-month graph(February-March) , both train and test set. Initially, we balance our data using random undersampling and thus, deleting a number of normal users, in order to be equal to suspended users. We then, select the significant features extracted from feature selection. In order to normalize our data before training in a range of 0 to 1, we utilize Min-Max scaler. We then store the model and the scaler, in order to be reused in the next steps.

To measure the initial performance of the trained model, we tried to predict user suspension on users that had social relations on each one-month period separately. Therefore, for our first scenario, after training our model on users of period 23

February to 23 March, we predict user account suspension in the following periods: 24 March - 23 April, 24 April - 23 May and 24 May - 23 June.

To predict user suspension for period 23 March - 23 April, we remove from the second graph (February-April) all users from the first graph (February-March). Respectively, to predict user suspension for the period 23 April - 23 May, we remove from the third graph (February-May) all users from the first graph(February-March) and the second graph (February-April). This prediction premise the balance of our model input. Before fitting our input for prediction, we transform the data with the scaler of the previous step.

6.2 Second Scenario

As seen in Table 6.1, we observe a spike on total user number in the period 23 February - 23 March, due to the fact that this period was the starts and the peak of the Russo-Ukraine war. After 24 March we observe that the number of users that had social relations rapidly decreased. At this step, should be highlighted, that graphs embeddings of the same users, change throughout the months. During Pytorch-BigGraph model training, each users' graph embeddings are positioned in dimensional space, relative to other users. Due to the addition of new users in each month's graph dataset, the embeddings change. For example, let's assume user A that appears on the February-March graph has graph embeddings [x, y, z]. For the same user A on the February-April graph, A's graph embeddings are [p, q, t], due to the fact that new users have been added. Respectively, user A's embeddings change on graph datasets of February-May and February-June.

For our second scenario, we want to discover how the model improves, throughout the period 23 February to 23 March. Using the above fact that the same user embeddings changes through time, the idea behind the second scenario is to train each month's user graph embeddings, according to users of the first-month (23 February - 23 March), since first-month contains more users, than any other month, due to the peak of Russo-Ukraine war.

Initially, we train our model with the best parameters from fine-tuning, utilizing as the train set the entire February-March graph embeddings dataset. Let's assume "U" is the list of user ids that are used for training. Therefore, for each graph embeddings dataset, we assume the following entities:

- previous users: users "U" that appear in February-March dataset.
- new users: new users added in the current graph embeddings dataset.

For example, in the February-April graph embeddings dataset, previous users will be users that appear in February-March dataset, and new users are the users that exist in the period from 24 March - 23 April. We, then fit our model with the modified embeddings of previous users, and predict user account suspension on new users. Before fitting our data model, we balance both the train and test set. For the purpose of normalization, we fit-transform MixMax scaler with previous

Scenario	Period	F1-score
1_{st}	24 March - 23 April	0.07
	24 April - 23 May	0.09
	24 May - 23 June	0.12
2_{nd}	24 March - 23 April	0.68
	24 April - 23 May	0.69
	24 May - 23 June	0.62
3_{rd}	February - March	0.72
	February - April	0.73
	February - May	0.75
	February - June	0.74

Table 6.2: Overall performance of all scenarios.

users and store the scaler. Before prediction, we transform new user embeddings with the loaded scaler, in order for the embeddings to be distributed on the same limits. We repeat the same process for each graph dataset.

6.3 Third Scenario

For our third scenario, we try to measure the performance of our model across all graph embedding datasets. For this purpose, we calculate the accuracy of the model, based the prediction of users' account suspension. What stimulated our interest in this scenario is how accuracy fluctuates throughout the 4-month period, considering the dimensions of each dataset. Be mindful of the fact that each graph, contains all previous graphs. Below we analyze the methodology we follow for this scenario.

For our first step, similar to previous scenarios, we balance our datasets with random undersampling. Due to an excessive number of users, we extract a number of normal users so as to match the number of suspended users. In order to preserve our model's generalization ability, we split each dataset into train and test sets, since the model fit on the train set and the test set remain hidden till final evaluation. Along these lines, the computational cost of the training is reduced and our model is generalized to predict new data that are hidden during training. An appropriate fit for our train/test split ratio seems to be 80/20. After the split, we normalize our train data utilizing MinMax scaler, and we store the scaler in order to be reused for the test set. Before prediction, we load the scaler, and transform the test set, in order to be normalized on the limits of the train set.

Chapter 7

Conclusions

With the integration of the above scenarios, we conclude with the following outcomes. From Table 6.2 we can observe that models in our initial scenario are not able to learn and adapt to time changes between social relations and follow the outcome changes of graph embedding. In comparison with the second scenario, where the model is able to achieve decent performance. The only problem with this approach is that users that are used at the initial selection (training set) may not be active after some time period, since Twitter may suspend malicious accounts. In this case, the model would be trained on outdated user relations that are not active and the model will produce a noisy outcome, as we can see in the performance of the last time period. In comparison with the two previous scenarios the last one, manage not only to adapt for dynamic changes of user relations but also those changes and growth of social graph are able to increase the model performance. Based on this research we manage to analyze multiple possible scenarios of malicious account detection based only on social relations. This work showed that social graph evolution during multiple months is required a specific approach in order to achieve maximum malicious account performance.

Bibliography

[1] Twitter batch compliance. `https://developer.twitter.com/en/api/compliance/batch-compliance/introduction`.

[2] Amit A Amleshwaram, Narasimha Reddy, Sandeep Yadav, Guofei Gu, and Chao Yang. Cats: Characterizing automation of twitter spammers. In *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–10. IEEE, 2013.

[3] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. Debot: Twitter bot detection via warped correlation. In *Icdm*, pages 817–822, 2016.

[4] Albert Chibuwe. Social media and elections in zimbabwe: Twitter war between pro-zanu-pf and pro-mdc-a netizens. *Communicatio: South African Journal of Communication Theory and Research*, 46(4):7–30, 2020.

[5] Farhan Asif Chowdhury, Lawrence Allen, Mohammad Yousuf, and Abdullah Mueen. On twitter purge: a retrospective analysis of suspended users. In *Companion proceedings of the web conference 2020*, pages 371–378, 2020.

[6] Farhan Asif Chowdhury, Dheeman Saha, Md Rashidul Hasan, Koustuv Saha, and Abdullah Mueen. Examining factors associated with twitter account suspension following the 2020 us presidential election. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 607–612, 2021.

[7] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 512–515, 2017.

[8] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web*, pages 273–274, 2016.

[9] Phillip George Efthimion, Scott Payne, and Nicholas Proferes. Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Science Review*, 1(2):5, 2018.

[10] Emilio Ferrara, Herbert Chang, Emily Chen, Goran Muric, and Jaimin Patel. Characterizing social media manipulation in the 2020 us presidential election. *First Monday*, 2020.

[11] Krzysztof Fiok, Waldemar Karwowski, Edgar Gutierrez, and Maciej Wilamowski. Analysis of sentiment in tweets addressed to a single domain-specific twitter account: Comparison of model performance and explainability of predictions. *Expert Systems with Applications*, 186:115771, 2021.

[12] Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[13] Clayton J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eytan Adar, Paul Resnick, Munmun De Choudhury, Bernie Hogan, and Alice H. Oh, editors, *ICWSM*. The AAAI Press, 2014.

[14] Ilia Karpov and Ekaterina Glazkova. Detecting automatically managed accounts in online social networks: Graph embeddings approach. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 11–21. Springer, 2020.

[15] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large scale graph embedding system. *Proceedings of Machine Learning and Systems*, 1:120–131, 2019.

[16] Muhammad Ali Masood and Rabeeh Ayaz Abbasi. Using graph embedding and machine learning to identify rebels on twitter. *Journal of Informetrics*, 15(1):101121, 2021.

[17] Igor Mozetič, Miha Grčar, and Jasmina Smailović. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036, 2016.

[18] Preslav Nakov, Vibha Nayak, Kyle Dent, Ameya Bhatawdekar, Sheikh Muhammad Sarwar, Momchil Hardalov, Yoan Dinkov, Dimitrina Zlatkova, Guillaume Bouchard, and Isabelle Augenstein. Detecting abusive language on online platforms: A critical analysis. *arXiv preprint arXiv:2103.00153*, 2021.

[19] E Puraivan, E Godoy, F Riquelme, and R Salas. Fake news detection on twitter using a data mining framework based on explainable machine learning techniques. 2021.

[20] Julio CS Reis, André Correia, Fabricio Murai, Adriano Veloso, and Fabrício Benevenuto. Explainable machine learning for fake news detection. In *Proceedings of the 10th ACM conference on web science*, pages 17–26, 2019.

[21] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*, 2018.

[22] Alexander Shevtsov, Christos Tzagkarakis, Despoina Antonakaki, and Sotiris Ioannidis. Identification of twitter bots based on an explainable machine learning framework: The us 2020 elections case study. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 956–967, 2022.

[23] Kirill Skorniakov, Denis Turdakov, and Andrey Zhabotinsky. Make social networks clean again: Graph embedding and stacking classifiers for bot detection. In *CIKM Workshops*, 2018.

[24] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 243–258, 2011.

[25] Cagri Toraman, Furkan Şahinuç, and Eyup Halit Yilmaz. Blacklivesmatter 2020: An analysis of deleted and suspended users in twitter. In *14th ACM Web Science Conference 2022*, pages 290–295, 2022.

[26] Svitlana Volkova and Eric Bell. Identifying effective signals to predict deleted and suspended accounts on twitter across languages. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 290–298, 2017.

[27] Xianchao Zhang, Shaoping Zhu, and Wenxin Liang. Detecting spam and promoting campaigns in the twitter social network. In *2012 IEEE 12th international conference on data mining*, pages 1194–1199. IEEE, 2012.