University of Passau
Faculty of Computer Science and Mathematics
Chair Software Engineering II

Master Thesis in Informatics

Towards the detection of malicious bots in Russian social networks

submitted by

Valeriia Stromtcova

1. Examiner: Prof. Dr. Gordon Fraser
2. Examiner: Prof. Dr. Christian Hammer
Supervisor: Isabella Graßl
Date: March 9, 2023

Contents

Abstract

This thesis concerns automatic detection of malicious political bots in Russian social networks on the example of VKontakte discussions around the Russian-Ukrainian armed conflict of 2022. This study attempts to close several gaps in the existing research around social bots. Firstly, it is one of the few studies investigating bot activity on an example of content produced in languages other than English. Secondly, it explores the effect political bots have internally in Russian-speaking communities and not the influence they produce on Western society. Thirdly, it presents a bot detection web tool available to the public that has no alternatives in the Russian-speaking segment of the Internet.

The thesis aims to build a model capable of detecting malicious political bots on VKontakte and explore the influence these bots produce on the online environment. The model is applied to VKontakte data in order to uncover political manipulations and capture the influence that bots have on this platform. We explore bots' influence in terms of the influentialness and sentiments of the user network. The results of the model can benefit both society and business because they all profit from increased transparency and level of trust in a social network as a result of the timely removal of bots and bot-produced content. We make the model results accessible to a broad audience on the Internet by building a web application "Bot-Checker". This allows anyone to see whether a VKontakte user is a bot or a real human.

The accuracy of the bot detection model, estimated on a subset of the data, reaches 85%. However, finding ways to confidently identify ground truth for bot-human classification and build other reliable bot detection models remains a crucial task for future research endeavours.

Acknowledgments

I want to express my gratitude to my supervisor from the University of Passau, Isabella Graßl, who provided invaluable feedback and assisted me in every possible way to ensure my success with the thesis. Isabella's professional advice helped me find the right means to conduct this study, and her friendly attitude was always a source of inspiration and optimism.

I could not have undertaken this journey without my husband Vitalii, who always believes in my success and does not cease to see the best in me. He is my greatest emotional support in these uncertain times. I am grateful to this wonderful person for always being there for me and teaching me perseverance, honesty and determination, without which any kind of scientific research would be meaningless and futile.

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The presence of autonomous agents in online social networks (OSNs), or so-called social bots, dates to the rise of OSNs and has long been an interest to computer scientists. Social bots are widespread: for example, according to Twitter's statistics, 15% of users on this social network are bots [Wis]. Of these, one-third are "malicious bots" created to misinform other users and manipulate their opinions [Wis]. The other two-thirds are identified as benign or "good" bots that do not serve malicious purposes.

Bots often actively participate in the online social discourse, generating an impact on the life of society. Probably one of the reasons why bots influence our opinions is how credible, sometimes indistinguishable from humans, their online behaviour is. Technological advances made their posts and even appearance very realistic. With the usage of such deep learning models as GPT-3[1], bots can authentically imitate human writing style; using realistic image generators, e.g., DALLE-2[2], they can pretend to have human faces. Moreover, ordinary users are sometimes unaware of bots on the Internet and believe that all the information they see on social networks is human-produced. This effect causes users to have an inadequate high level of trust in the content on social networks.

It is not only people for whom the distinction between bots and real users is a challenge. Automatic methods, including ML models, sometimes achieve very high accuracy of bot detection on test datasets. However, when released "in the wild", they fail to function with the same efficiency [Cre20]. Moreover, bot developers constantly create more and more sophisticated bot accounts, and it takes some time for researchers and practitioners to adapt their bot detection models for these new kinds of bots. As a result, many social bots remain unnoticed.

Apart from adjusting the bot detection models to new generations of bots, another challenge for this research area is building models able to process content in languages other than English. Currently, there is not much research on bots that produce content in another language, e.g., Russian. The query "bot detection" AND "English language" yields seven times more search results in Google Scholar than the same query with "Russian language".

Meanwhile, current political events in Russia and the entire world (the current Russian-Ukrainian armed conflict that escalated on the 24th of February 2022) influence worldwide society in many ways. This conflict is widely discussed in OSNs, with conflict-related posts published and numerous comments appearing under them every day. Therefore, it is crucial to understand what role social networks play in these events and, more specifically, how bots manipulate people's opinions on all sides of the conflict.

[1] https://beta.openai.com/docs/introduction/overview
[2] https://openai.com/dall-e-2/

The Russian-Ukrainian armed conflict of 2022[3] is also known as the Russian-Ukrainian war, the Russian invasion of Ukraine, or the special military operation of Russia in Ukraine. Since all these terms are associated with a certain attitude towards the events of the conflict, be it a Pro-Russian or Pro-Ukrainian position, we will adhere to the most neutral term "armed conflict" to avoid bias and subjectivity.

This conflict is a severe escalation that started on 24th February 2022 after a long confrontation between countries in the regions of Donetsk, Luhansk and Crimea, during which, according to The Office of the United Nations High Commissioner for Human Rights (OHCHR), at least 3,400 civilians were killed [Sta22b]. On the morning on 24th February 2022, the president of Russia, Vladimir Putin, declared this escalation, motivating it with "denazification" and "demilitarisation" of Ukraine, preventing the expansion of the North Atlantic Treaty Organization (NATO) and denial of the very idea of a separate Ukrainian identity and the legitimacy of the Ukrainian state [Man22]. Minutes after the speech, missiles hit numerous cities across Ukraine. Aside from the dramatic numbers of killed and injured civilians (8,173 civilians killed and 13,620 wounded, by the August 2022 estimations by OHCHR [Sta22a], the conflict "triggered a tsunami that dramatically impacted the world economy, geopolitics, and food security" [Per+22]. Moreover, there is a tremendous impact on human health and the environment [Per+22]. Such an impactful global event presents an essential target for scientific research. Now, more than a year after the escalation of the conflict, the query "Russian-Ukrainian armed conflict 2022" in Google Scholar returns over 28,200 papers on the subject. All these studies approach the conflict from a different perspective: economics, geopolitics, history, ecology, medicine, etc. In the upcoming years, we will most likely observe a boom in scientific interest in this research area.

The Russian-Ukrainian armed conflict is undoubtedly one of the most popular topics on social networks. Especially at the beginning of the escalation, numerous accounts on Twitter, Instagram, Facebook, VKontakte and other OSNs were buzzing with discussion, with thousands of posts and comments published every day. The confrontation between pro-Ukrainian and pro-Russian OSN users was heated and emotional right from the start. It was not rare to see social network users claiming that someone they disagree with is a social bot. However, nobody was able to precisely identify bots participating in this discourse.

Thus, the motivation of the current study is to take a look at the conflict from the perspective of the intersection of sociology and computer science and try to uncover the political bots that aim to manipulate the minds of Russian-language social network users. Developing a new political bot detection tool can help researchers and society better understand the dynamics of the ongoing conflict, its impact on the online environment and, ultimately, on the real people who use Russian-language social networks.

Aside from being a relevant research area for society during major political events, bot detection presents a crucial business need for companies that own social networks. The presence of bots in these networks can undermine user trust and threaten a company's image. Moreover, as the recent scandal related to Elon Musk's possible acquisition of Twitter shows [Zah22], bots can even become an obstacle to 44 billion dollar deals. Finding and eliminating bots presents a task of increased interest for OSN owners. Therefore, bot detection is a relevant task both for businesses and society.

[3] https://www.cfr.org/global-conflict-tracker/conflict/conflict-ukraine

This paper aims to develop a method to automatically detect bots that produce content related to the Russian-Ukrainian conflict of 2022 in the Russian-language social network VKontakte. Moreover, the goal is to analyse their behaviour and understand common patterns; figure out their role and influence on the OSN landscape; provide the audience of the social networks with a tool to check if a given user is a bot or not, in order to increase awareness of the existence of bots in OSNs.

The contributions of the thesis include:

a) Collection of posts and comments dataset from VKontakte;

b) Development of three bot detection models based on this dataset;

c) Analysis and evaluation of the results of the bot detection models;

d) Identification and analysis of the influence that bots have on the discourse on VKontakte;

e) Development of a web interface that allows checking if a user is identified as a bot or a real human.

1.2 Research Questions

The goal of this research is to explore the bot landscape on the social network VKontakte in relation to the escalation of the Russian-Ukrainian armed conflict in February 2022. In order to do so, the first step is the collection of a dataset of content with the usage of the VKontakte API. Then, a bot detection model based on the recent advances in this area should be developed. Evaluation of the model's results should also be performed. After that, the next task is to identify the influence of bots on the network. To do so, several techniques are applied, including sentiment analysis.

Another contribution of this project is to develop a tool to let social network users check other users' accounts and automatically classify them into bot- and non-bot accounts. The end goal of such a tool would be to raise awareness of social bots in the Russian-speaking community and attract users' attention to the problem of misinformation spread in OSNs. The tool should be publicly available on the Web and function automatically on the base of the model developed in the course of this research.

The thesis research aims to detect the bots that post information about the Russian-Ukrainian armed conflict of 2022 in OSNs; analyse their behaviour and understand common patterns; figure out their role and influence on the OSN landscape.

The thesis research attempts to answer the following questions:

a) How to automatically detect bot accounts that spread propaganda on the topic of the Russian-Ukrainian armed conflict of 2022 on Russian social networks?

b) What influence do these bots have on the VKontakte discussion around the aforementioned conflict?

c) How to make bot detection results available to the public?

2 Background

2.1 Definitions

To introduce the reader to the context of this research, here are the key definitions of terms used throughout this paper. The definitions are either directly cited or rephrased from existing studies. Since a large body of work on social bot detection already exists, the definitions should be carefully picked from the most cited sources in top Computer Science journals. Higher citation scores usually indicate scientists' trust in a research paper and can be suggestive of the higher quality and importance of a study. Setting a citation threshold to 100 for the bot-related studies, we eliminate the papers that have been cited less than 100 times and thus adhere to more well-known and agreed-upon definitions. However, since modern political events have not yet been described in numerous studies, and information about them is contained in recent papers that do not have a high citation count, no citation threshold is set for them.

Online Social Networks (OSNs), also known as social network sites (SNSs), are "web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system" [BE07]. With the current development of technology, OSNs can be not only web-based but also mobile-based. However, there is no newer definition that would include the "mobile" aspect of OSNs. So, we will use the existing definition but supplement it with the "mobile" aspect: "OSNs are web- or mobile-based services..." Examples of Online Social Networks include, e.g., Facebook, Twitter, and Instagram; for CIS countries, this list is extended with such platforms as VKontakte and Odnoklassniki.

In OSNs, real users co-exist with automated accounts (social bots). Currently, there is no well-agreed definition of what a Social Bot is [Cre20]. Social bot research exists at the intersection of sociology and computer science and is looked at from different perspectives, e.g., social scientists' perspective or engineering perspective. Some definitions focus on technical details, and some highlight social interaction. However, efforts were made to unify the terminology and outline a common definition for this term. For instance, in [Gri+17], the following definition is provided: "The term 'Social Bot' is a superordinate concept which summarises different types of (semi-) automatic agents. These agents are designed to fulfil a specific purpose by means of one- or many-sided communication in online media". Throughout this thesis, we will adhere to this high-level definition as the most universal and comprehensive.

Social bots have multiple purposes, depending on their creators' goals. One of the types of social bots is Political Bot. This bot type is relevant to the current research, as this study aims to uncover the bots that attempt to influence the political landscape of VKontakte. As stated in [Woo16], Political Bots are politicised social bots that "suppress free expression and civic

innovation through the demobilisation of activist groups and the suffocation of democratic free speech. They subtly work to manipulate public opinion by giving false impressions of candidate popularity, regime strength and international relations." Bots of this type usually express higher activity during important political events, such as elections, political crises, and conflicts.

2.2 Related work

Bot detection in general is an established field of research that has been of interest to scientists for over a decade. Researchers regularly publish papers regarding bot detection in the top Computer Science journals and introduce new methods to detect bots based mostly on the recent advances in Deep Learning. When bot detection was yet an emerging area (2010), scientists mostly used supervised machine learning methods that focused on individual accounts and classified them into two categories: a bot or a real user. Research shows that, as bots were becoming more and more sophisticated, the naïve assumptions of early supervised methods ceased to function with the same efficiency [Cre20]. Instead, in 2012-2013, new methods for bot detection based on the so-called group approach emerged. In group methods, "a detector analyzes a group of accounts, looking for traces of coordinated and synchronized behaviours. Large groups of coordinated accounts are more likely to be detected than sophisticated individual bots" [Cre20]. The reason for applying group-based methods is the assumption that bots behave in a coordinated manner more often than human users. Since 2018, most of the published papers in the field of bot detection have made use of group approaches (e.g. [Cre+20]). The majority of new models still use the supervised learning approach ([Cre+20]), which assumes that ground truth is known and a model can learn based on this ground truth. However, in reality, the verifiability of the ground truth in such research is questionable: labelled data comes from crowdsourcing, meaning that humans label this data. At the same time, humans, according to recent data, are only able to correctly classify bots in 24% of cases [Cre20]. Therefore, unsupervised learning has gained more and more attention in the recent years. Simultaneously, adversarial machine learning methods employing anomaly and peak detection are also on the rise.

A narrower and less popular field of research among computer scientists is bot detection on Russian-language social networks. Only a few papers have been published in recent years regarding this topic, and we were not able to find any of these papers published in major Computer Science journals, e.g. proceedings of ACM, AAAI, ICML or IEEE. Speaking about the topical coverage, the majority of papers published in this area relate to the influence that Russian social bots have on the Western audience. As an example, a recent paper [Rhe21] explores the impact of social bots on electoral campaigns (primarily the Western ones) and only mentions Russia in the context of interfering with the US 2016 presidential elections. An extensive study of Russian propaganda in Eastern Europe briefly touches on the topic of bots, stating that in 2015, 17,590 accounts on Twitter could be classified as pro-Russian social bots [Hel+18]. Another paper dedicated to measuring the political orientation of Twitter bots in Russia aims to discover the ratio between pro-Kremlin and pro-opposition bots and analyses the primary topics of the content they produce [Stu+19].

Most papers that aim to detect bots in Russian social networks use datasets collected from

Twitter. One reason for this could be that Twitter is a social media to which researchers outside Russia have easier access and with which they are more familiar than with originally Russian OSNs. However, taking a look at the statistics of the usage of different social media by the Russian population [Sta21], it can be seen that Twitter only has a penetration rate of 11,7%. In the meanwhile, VKontakte, a Russian social network platform, boasts a gigantic penetration rate of 76,4%. Therefore it seems promising to make use of the data offered by this media platform. Surprisingly, to our knowledge, few papers cover bot detection on this platform. An example of bot detection research on VKontakte is [VLR19], where the authors present their findings regarding topical coverage of bot-produced content on VKontakte but do not provide enough details for their bot detection model to be reproducible and easily checked.

Overall, the current state of research indicates that even though bot detection is a long-studied area, it still has several unsolved problems. One of them is that old bot detection techniques (based on supervised learning and individual consideration of each user account) became obsolete with the development of more sophisticated bots. There is a need for new bot detection models. These new models draw primarily from the group-based approach and either supervised, unsupervised or adversarial learning. Another challenge is that few scientists research bot detection on the examples of content produced in languages other than English, e.g. Russian. Moreover, even when a paper focuses on Russian content, it still centres around the West-oriented approach, e.g. studying the effect of pro-Kremlin bots on Western society. A large gap exists in the current research, and the recent political events call for closing this gap.

3 Methods

3.1 Collection of data from social networks

3.1.1 Initial dataset

This study relies on a dataset called VoynaSlov that has been previously collected in [Par+22]. This dataset consists of more than 21 million Russian-language social network activities, such as tweets, posts and comments. The data is taken from two major OSNs: Twitter and Vkontakte. The authors of the dataset have identified the hashtags for Twitter and Vkontakte that are related to the Russian-Ukrainian conflict and have run a search for social media posts and comments under them. The social media used in this dataset can be classified as either state-affiliated or independent. For the complete list of hashtags and social media used, see the original paper [Par+22].

Only a subset of this dataset is of interest for this study, specifically, the 5,6 million comments made by VKontakte users after 24.02.2022. Since the dataset only contains comment identifiers and not the whole comment metadata and content, the dataset should be enriched with additional data from VKontakte. Before doing that, choosing the right tools to do so is necessary. Therefore, the following section describes the choice of development tools.

3.1.2 Choice of tools

To implement the data collection, the Python programming language is used. Version 3.10 of this language is the latest stable version and is applied throughout the project. Python is characterised by its simplicity and a vast array of additional tools, such as libraries and frameworks, that allow extending its functionality. Moreover, Python is widely used in scientific research.

Dependency management is performed with a virtual environment (venv). The project uses a new virtual environment where all the libraries and frameworks are installed. This enables a convenient separation of the project dependencies from the rest of the packages installed on the same machine in different environments.

To collect data from the social network VKontakte, the open Application Programming Interface (VK API)[1] provided by this platform was used. To use this API, it is necessary to have an account on VKontakte and create an app under this account. Then, a developer should obtain an authentication token later used in all requests sent to the API.

VK API has several peculiarities that make working with it more difficult. Firstly, sending more than three requests per second with the same authentication token is not allowed.

[1] https://dev.vk.com/api/getting-started

Secondly, there is a restriction on the total number of requests over a period of time. VK does not provide any exact numbers for the second restriction, but it was not an obstacle to the current research. The VK library for Python[2] provides a wrapper over VK API methods and ensures easy and convenient access to VKontakte data. Therefore, this library was used throughout this project.

Having fetched the data from VK API, it should be stored in a database for further processing. Therefore, MongoDB is another crucial tool for this research. Mongo is a NoSQL cross-platform document-oriented database that utilises JSON-like documents. It makes use of the concepts of "collections" and "documents". Documents are the basic database entries, while collections represent lists of documents. Since the data received from VK API is in JSON format, inserting it into a MongoDB database is effortless and very convenient. No data format transformers are needed to transfer the data from VK API responses to a MongoDB storage. The pymongo library used in this study provides tools to interact with MongoDB databases from Python programs.

Another development tool used in this project is Docker. It employs virtualisation to deliver software in packages (so-called containers). Docker is used in this study to easily deploy the program to a remote server (described later in section 3.1.4).

During the implementation of the data collection step, several secret values were used — for example, the database password or VKontakte authentication token. These secret values should not be published in the project repository to ensure the security of these values and forbid access of external viewers to the research data. A common practice in such cases is to use a .env file in the project and store the values there. Then, a library such as dotenv allows a Python program to access these values easily. The version control system ignores the .env file because its name is written in the .gitignore file. The .env file is only stored locally on the developer's machine. When deployed to a server, these secret values are taken from the server's "config vars" that are only available under the specific account. This approach allows for securely storing secret values, such as passwords and keys, without exposing them to external viewers. Thus, it prevents undesirable access to the data collected throughout the research and any modifications by third-party developers.

Applying all the aforementioned tools allows for a robust and convenient approach to developing the data collection program. In the next section, the development of the program itself is outlined.

3.1.3 Implementation of the data collection program

The data collection program consists of three components.

The first component is the main. It is used as an entry point for launching the data collection process. It uses the rest of the components (data_parser and database_adapter) and ensures that the data received from VKontakte is stored in the database.

The database_adapter component is dedicated to any database communication, such as retrieving data based on a query, inserting a new document, or updating an existing one. One of the functions of the database_adapter component is the initial population of the database

[2] https://pypi.org/project/vk/

with comment identifiers from the voynaSlov repository. This function takes the comment IDs from the repository and creates a document in the database for each comment. At this first stage, each document only consists of five fields:

a) _id, the internal identifier of a document in Mongo;

b) vk_id, the VKontakte identifier of this comment;

c) media_name, the name of the media under which this comment is left;

d) media_id, the VKontakte identifier of this media;

e) processed, a boolean feature identifying if this comment has already been enriched with all the required features or not. This field is set to False for all comments on this stage.

After this function processes all the comment IDs and creates all the necessary initial comment documents in the database, the data_parser component should be used. It contains the code required to enrich every comment with additional features from VKontakte. VK API is used to implement such an enrichment. The method wall.getComment[3] is used to get the extended information about each comment by its vk_id. This method also returns information about the users who left the comments, which is convenient for extracting the user features from this data later. However, the VK API on its own is not enough to retrieve all the user information needed for further analysis. Another source of data is the so-called FOAF request[4] to VKontakte that allows for parsing the missing user features, such as the date on which a user account has been created or the number of followers of this user. The complete list of features retrieved for each user is available in Appendix A.

At first, the naïve approach of sequentially running the wall.getComment method for each of the 5.6 million comments performed very poorly. Processing only three comments per second, this program would have taken more than 21 days to finish extracting the comment data. Therefore, two improvements were implemented to speed up the process:

a) we took advantage of the execute VK API method[5] that allows the inclusion of a sequence of up to 25 method calls into one single request.

b) The program was deployed to a remote server, and two parallel processes with separate VK API tokens were started there. Another process has been running on the local development machine.

Using these improvements, it became feasible to make the data collection perform more than 70 times faster than the initial naïve version of the program. Thus, the comment data enrichment phase only took several days, including the first few days when the program ran in a non-parallel, naïve way.

Finally, having parsed the enriched data for each comment, we again utilise the database_adapter component to update the initial comment documents with new fields and create documents for users.

Figure 3.1 presents the data collection program sequence diagram. All the steps of data processing described above are depicted in this diagram. Like this, we process all the comments

[3]https://dev.vk.com/method/wall.getComment
[4]https://vk.com/foaf.php
[5]https://dev.vk.com/method/execute

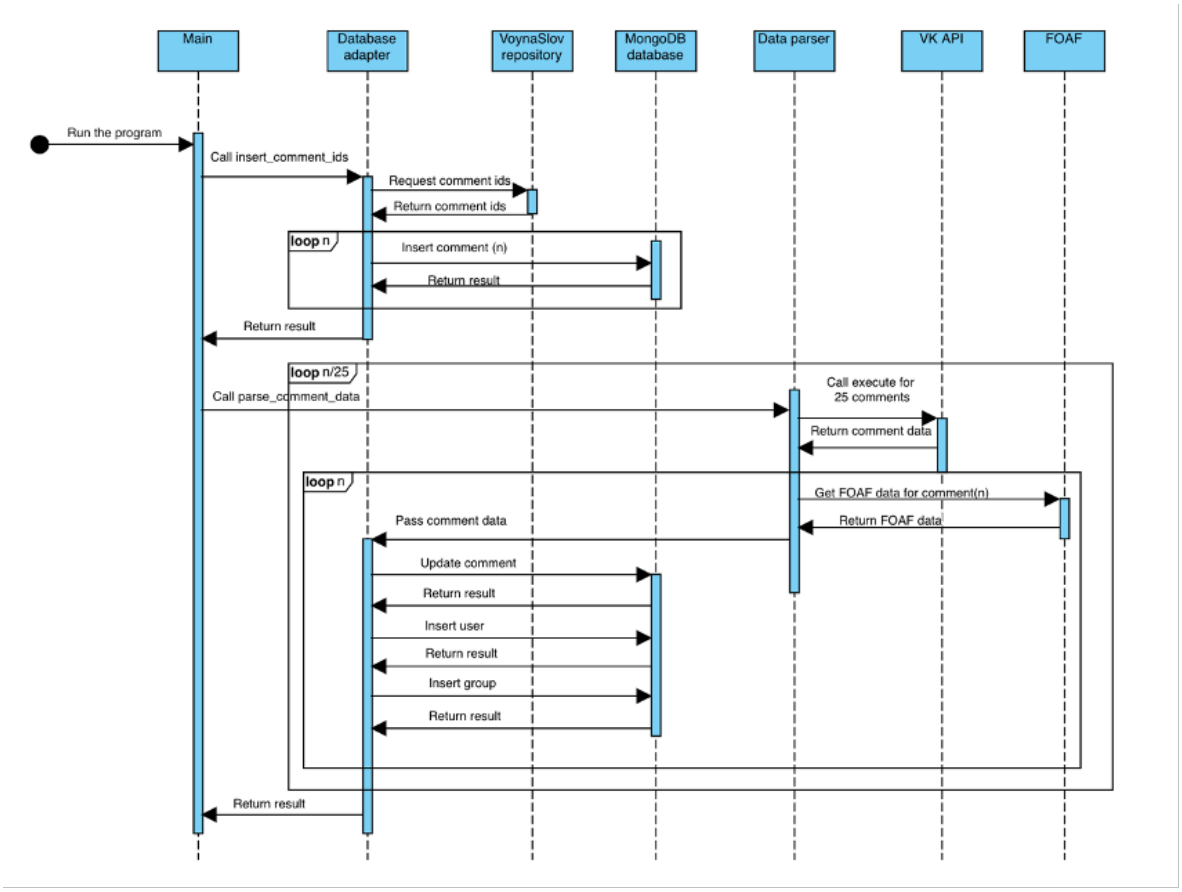

Figure 3.1: Sequence diagram for the data collection program

from the voynaSlov repository, extract all necessary information about them from VKontakte, and save this information in the database.

3.1.4 Deployment to the remote environment

The data collection program was first created and tested on the local development machine. However, as the program's database size and memory requirements grew, a need for a remote deployment environment appeared. The remote environment allows storing all the data and running the program on remote, cloud-based servers, which makes the data collection process independent of the memory and performance restrictions of the local machine.

A version control system (VCS) is necessary for this project, as it allows for easier management and distribution of the source code, including its delivery to a remote environment. Git, the most popular VCS, is used in this study. It is a free, open-source, fast and reliable system that allows branching, saving the history of modifications (commits) in the source files, and quickly switching between the branches or commits. Supplemented with Github, it also allows setting up the Continuous Integration and Delivery pipeline.

MongoDB provides a reliable cloud-based database system (MongoDB Cloud). A database with 10GB of allocated memory was created for this project. MongoDB Cloud allows the management of the database through a web interface, setting up database backups, and monitoring database metrics.

Heroku is a cloud-based platform allowing to deploy various applications. All time-consuming operations are deployed to and run on Heroku so that the data collection process does not depend on the state of the local machine. Heroku supports Docker containers, and the containers running on Heroku can access the MongoDB cloud.

In Figure 3.2, you can see the deployment diagram that depicts both local and remote development environments and the relations between them and the components inside.

From the local development machine, after being tested in a local virtual environment, the code is pushed to GitHub using Git. Along with the code, two configuration files are pushed: Dockerfile and Heroku.yml. The Dockerfile defines the configuration of a Docker container that should run on the Heroku server. The Heroku.yml file is a configuration file for Heroku that describes how Heroku should run the program. The code is deployed to a Heroku server from a repository on the GitHub server. A Docker container is built on this server, and all the components inside it start during the program's execution. During execution, the Database adapter component establishes a connection to the MongoDB Cloud server and makes queries, inserts and updates. The program continues to run until all comments from the initial dataset have been processed and extended information about them is stored in the database.

Running the program in a remote environment is an error-prone process. Errors can occur during the execution on the Heroku or MongoDB sides. Therefore, it is crucial to monitor the performance of the deployed program.

MongoDB Cloud and Heroku provide sufficient real-time opportunities to monitor programs and database operations execution.

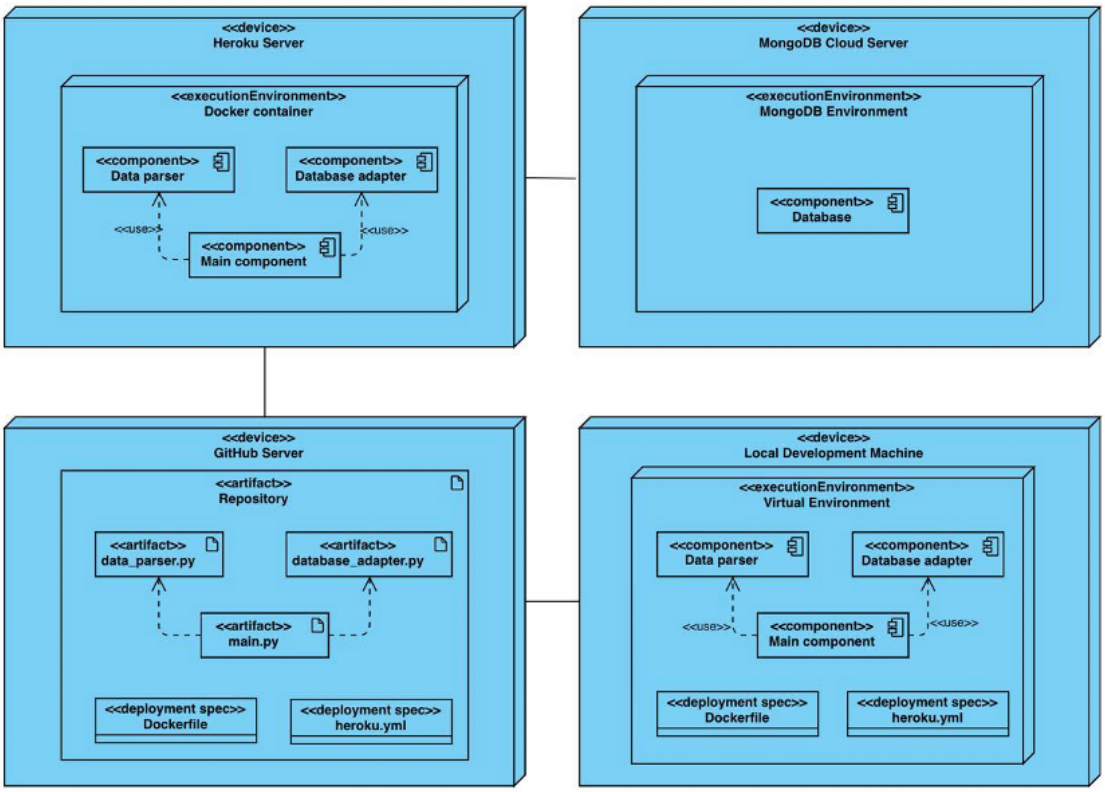

Figure 3.2: Deployment diagram

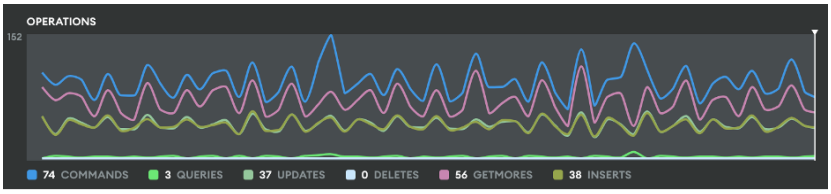

Figure 3.3: Real-time monitoring in MongoDB Cloud

Figure 3.4: Logs monitoring in Heroku

Figure 3.3 shows an example of real-time monitoring in MongoDB Cloud. Using this chart, it is possible to monitor the speed and number of database queries, updates, inserts and deletes. The two most important metrics in the data collection step were “Inserts” and “Updates”. During comment data parsing, we continuously update the comments collection in the database and insert documents into the “users” collection. Figure 3.3 shows a typical chart for unparalleled execution of these operations.

As seen in Figure 3.3, the speed of database updates and inserts fluctuated around 40 per second. However, with parallelisation into three separate processes (one running locally and two running on a Heroku server) with different API tokens to avoid VKontakte restrictions, it was possible to increase the speed of both updates and inserts to 120 operations per second.

Heroku logs present another opportunity to monitor the functioning of the program online. After a deployment, Heroku starts a “dyno” process in several seconds. This process can crash down but will be automatically restarted by Heroku. An example of such behaviour can be seen in Figure 3.4.

It is also noticeable that Heroku logs all the errors that occur during the execution of a program, which helps debug later.

3.1.5 Resulting dataset analysis

The resulting dataset contains data about:

- 5 599 287 comments;
- 967 groups;
- 283 506 users.

About 5 000 comments were not parsed from VKontakte because they had already been deleted from the platform by the time of data collection. Some users were also deleted or

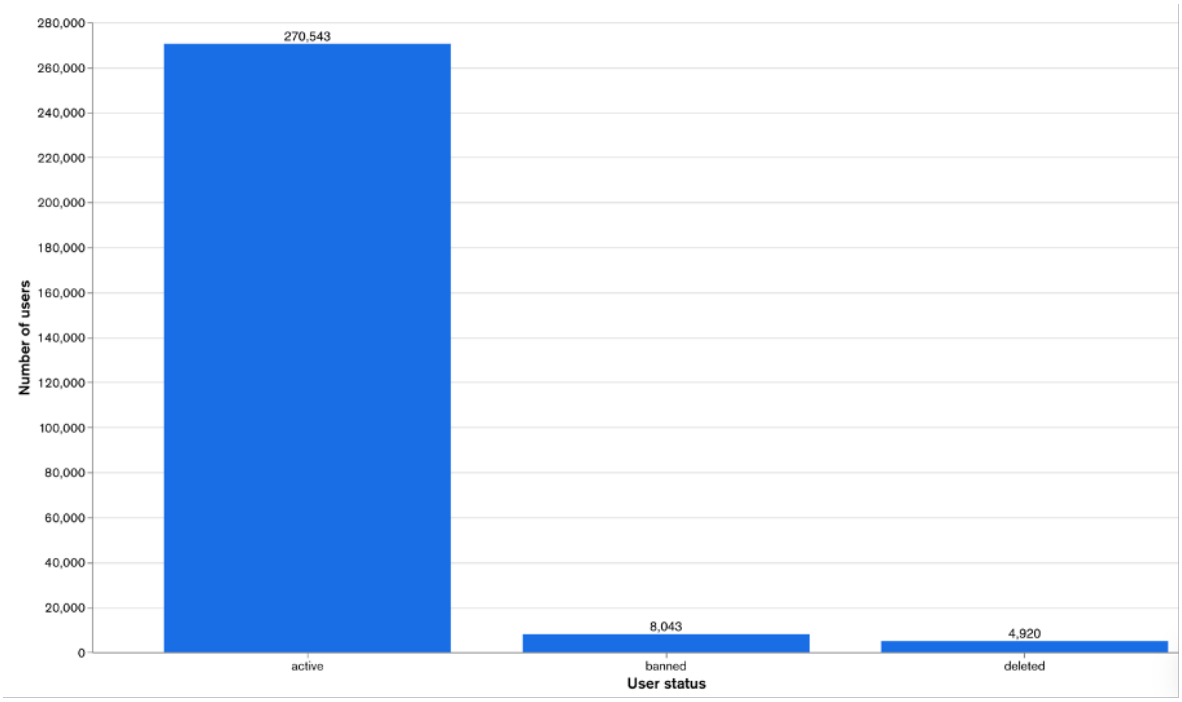

Figure 3.5: Distribution of users by user status

banned from the platform, but VKontakte stores data about them, so it was successfully parsed and stored in the database. There are 12 963 such users in the dataset. We keep all the data returned in the VK API response and some FOAF attributes for each user. There are 14 fields for each comment and 21 for each user in the database. These fields will become features for the bot detection models.

In order to successfully build bot detection models, it is necessary to conduct an exploratory analysis of the data collected. Taking a look at the dataset statistics allows us to gain insights into the data and make better use of the dataset for the model development, taking the dataset specifics into account. The exploratory data analysis was conducted on two document collections ("users" and "comments"). The charts below are created with MongoDB Charts[6], which allows building dashboards from data stored in a Mongo database.

The "users" collection contains documents concerning 283 506 VKontakte users. Each of these users has a particular status at the moment of data collection. Firstly, we explore the user statuses in the database. The distribution of users by user status is shown in Figure 3.5.

It can be observed that most users (95,5%) have an "active" status. Another 2,8% and 1,7% have a status of "banned" or "deleted", respectively. The "banned" status indicates that a particular user has been blocked by the VKontakte moderators. The reasons for such a block can be spamming, other fraudulent behaviour, or breaking the rules of the platform in any way. The "deleted" status indicates that a user deleted their account. The fact that most of the users are active on the platform is helpful for the current research, as the extended set of features from VKontakte can only be collected for active users. The data about any banned

[6]https://www.mongodb.com/products/charts

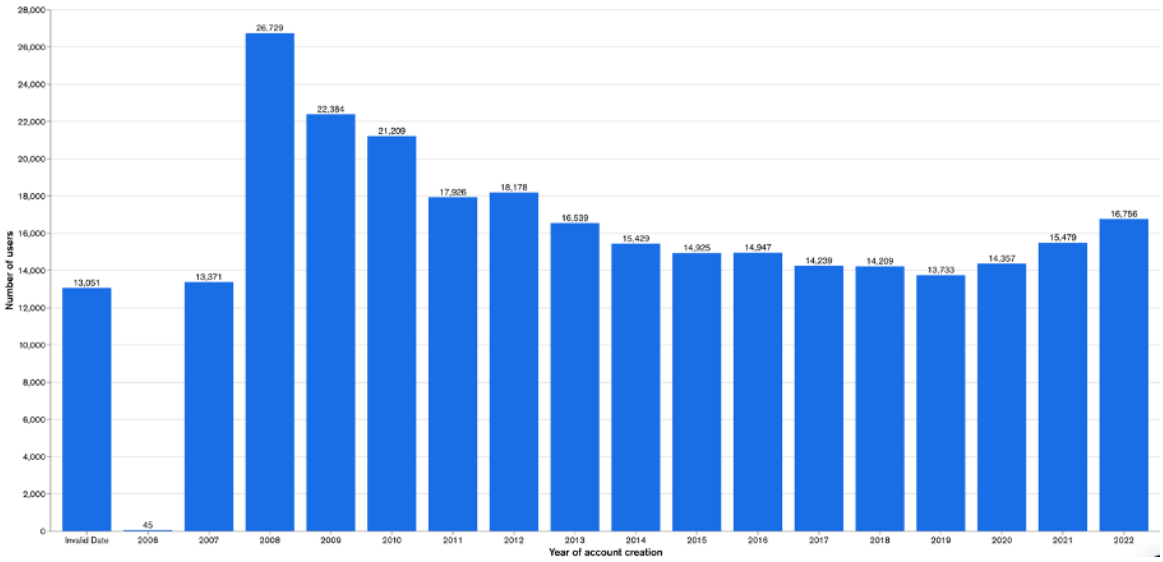

Figure 3.6: Distribution of users by year of their account creation

or deleted user is limited to a few fields, on which it would be hard to build a bot detection model.

Aside from the user statuses, it is also useful to explore the account creation dates of the users. We hypothesise that if there was a significant increase in the number of recently registered users, it could indicate a high number of new bots appearing on the platform. Figure 3.6 displays the distribution of users by year of account creation.

In Figure 3.6, an interesting pattern is observed. Since its foundation in 2006, VKontakte has attracted thousands of users each year. The peak of popularity for new users is the year 2008. During this year, 9,4% of the users in our dataset have signed up for VKontakte. Then, the number of new registrations tends to decline. However, starting from 2019, the numbers show a positive trend. In 2022, more than 16 000 users from our dataset have registered in VKontakte. Since we only collected the data about comments left until May 2022, and 2022 is not over yet, we can suppose that by the end of this year the number of new registrations will at least double and overcome the previous record set in 2008. Therefore, a significant peak in registrations is observed during the last year. Although it is not a clear indication of a new wave of bot accounts, it can be an indirect sign of such a trend. March was the most popular month for new registrations since the beginning of 2022. In March, the number of new users that entered our dataset grew three times higher than the February level. This is shown in Figure 3.7.

Further, it could be helpful to explore the commenting behaviour of users in the dataset. Calculating the user activity throughout the dataset, we create a comment_rate field in each user database entry. The distribution of the values in this field is shown in Figure 3.8.

In Figure 3.8, a threshold of 20 comments is applied in order to make the chart more comprehensive. All the numbers higher than 20 comments are not displayed. About 41,2% of the

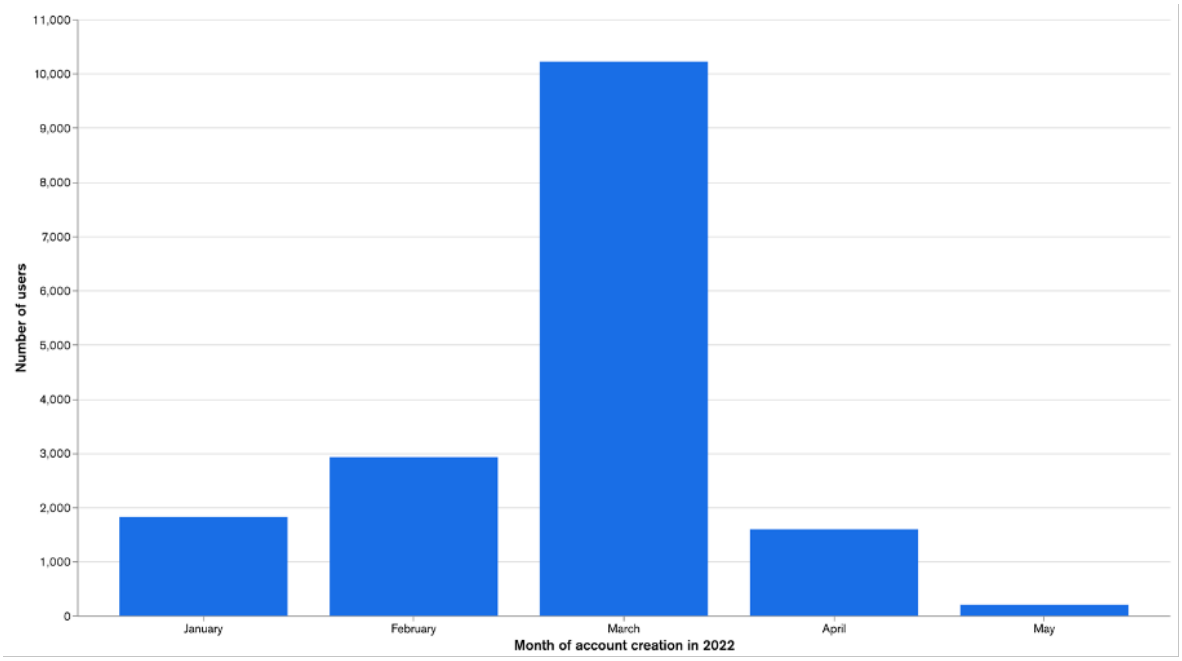

Figure 3.7: Distribution of users by month of their account creation in 2022

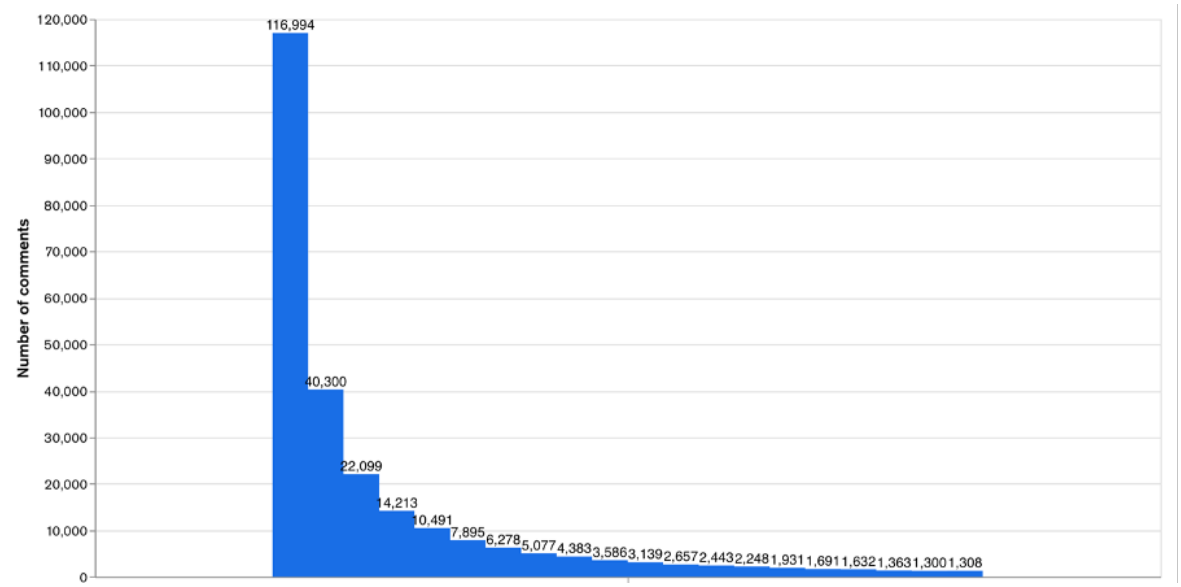

Figure 3.8: Number of comments per user

users only left one comment in the dataset. After the peak value at 1 comment per user, the distribution follows a declining exponent. Very few users left more than 8 comments in the dataset. This chart includes data about 251 028 users who left from 1 to 20 comments, with the remaining 32 478 users leaving more than 20 comments in the database. This behavioural peculiarity may be important for building the bot detection model.

As a conclusion regarding the "users" collection, most of the users in the dataset are active, and only 4,5% of the users are banned or deleted. Moreover, we explored when the user accounts were created. There has been a slow but steady rise in the number of accounts created since 2019. The data also shows a peak in new user registrations in March 2022, which possibly can be indicative of the appearance of new bots on VKontakte. Finally, we took a look at the behavioural activity and found out that most of the users in our dataset left only one comment. Interestingly enough, some users post thousands of comments. For example, the most active user has 5 836 comments in the dataset and displays an active pro-Russian position. The second most active user has left 4 925 comments and his attitude is completely opposite (pro-Ukrainian). However, these users are rare exceptions from the overall trend with most users leaving from one to three comments.

At this stage of the analysis, it is impossible to make solid conclusions and understand who of the users is a bot and who is a real human. However, the data exploration step provides insight into the collected dataset and allows us to identify common patterns between users and detect outliers.

The "comments" collection contains 5 599 287 documents. Firstly, the number of valid comments is checked. About 20% of comments in this collection are marked as invalid. This means these comments were deleted, and no data is available anymore. Several reasons can cause such a high number of invalid comments: for instance, the newly introduced Russian laws that prohibit the "discrediting of the Russian army" and censor social networks. Some users might have been afraid of being prosecuted for their comments and deleted them. Another reason could be the VKontakte moderator's fight against bots and the deletion of spammy comments.

The distribution of comments by month in the year 2022 is shown in Figure 3.9.

A visible spike is observed in March with over 1 900 000 comments during this month. April's numbers follow but are slightly lower. In May, the number of comments decreased significantly. The leftmost column marked as "Invalid date" represents invalid comments for which the data could not be parsed from VKontakte. It appears that VKontakte users have been talking a lot about the escalation of the Russian-Ukrainian conflict right after this escalation happened, and in March they were concerned about these events and ready for discussions in the comments. However, in May the commenting rate significantly decreased. Potentially, the users could have lost interest in the ongoing conflict, which could explain this decline.

Next, we are taking a look at the distribution of comments and posts over social media. Figure 3.10 displays this distribution in a column chart where the number of posts is marked in green colour, while the number of comments is plotted in blue. The short names of social media accounts are displayed on the X axis.

The most popular social media both by the number of posts and comments is RIA. It is a

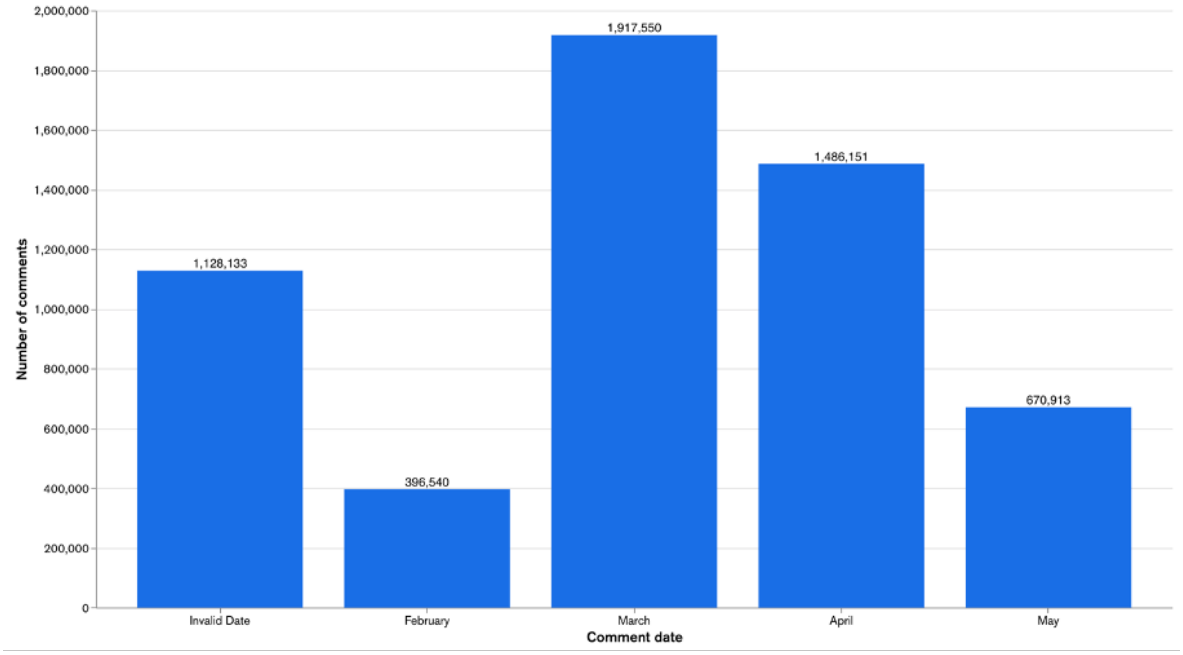

Figure 3.9: Distribution of comments by month

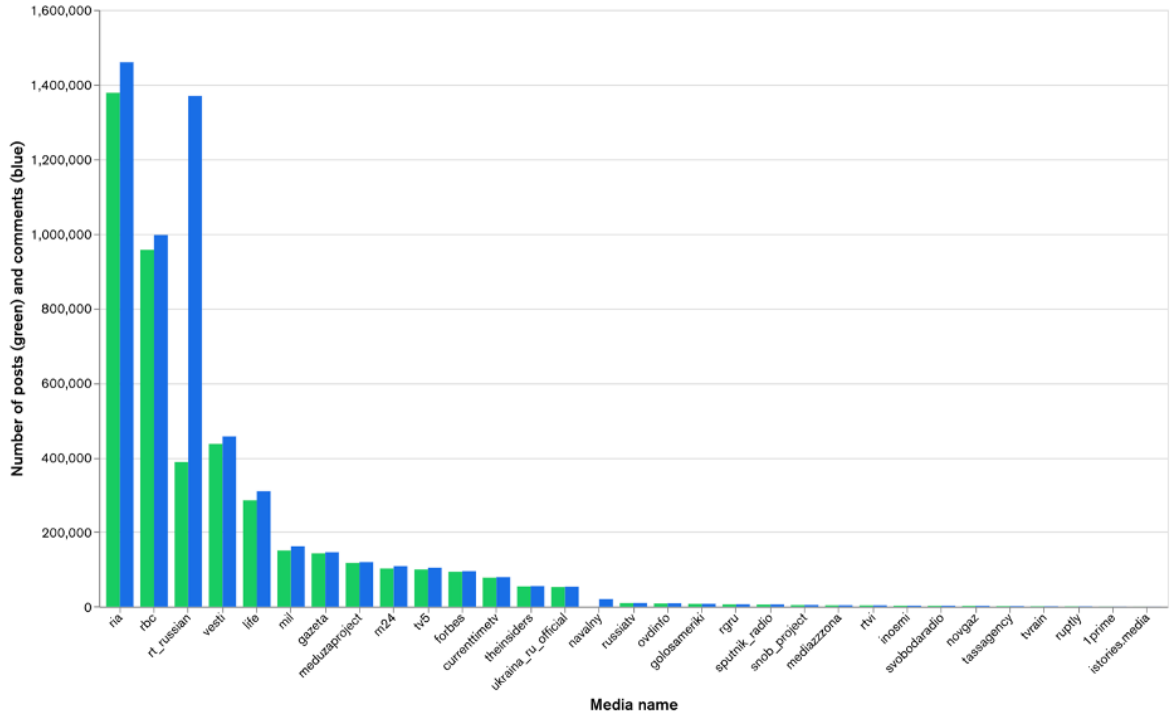

Figure 3.10: Distribution of posts (green) and comments (blue) by social media

state-affiliated social media, as well as RBC, RT Russian, Vesti, Life, MIL, and Gazeta, that occupy the first 6 places by the number of comments and posts. The first independent media in this list (Meduza) takes only 8th place. Overall, there is much more state-affiliated media content in the collected dataset than the content from independent media. For example, the dataset contains about 5,2 million comments on posts created by state-affiliated media and only 400 thousand comments on posts by independent media. The comment/post ratio is nearly the same (around 1,05:1) for all social media, whether state-affiliated or independent. However, RT Russian presents an exception to this rule. On average, there are 3,5 comments under each post on this social media. This anomaly is promising for further analysis and can potentially indicate the presence of bots on this social media.

3.2 Building the bot detection models

3.2.1 Absence of ground truth

Regardless of the chosen method of bot detection, the biggest challenge of this study is the absence of ground truth on which of the users is a bot. Different studies solve this problem in different ways, however, none of them is perfect. Therefore, any training of the model and evaluation of its performance is rough, approximate, and should be critically taken into account.

In most existing works where an unsupervised approach is applied, the results are evaluated and verified with the usage of existing tools such as Botometer that are believed to provide exact estimates for such models. However, Botometer does not work with VKontakte data. Moreover, Botometer is claimed by some researchers as an unreliable bot detection tool because of a high number of false positives [GK22]. Therefore, it is not possible to copy the evaluation strategy from existing work. In [ABS16], the authors provide three alternatives to Botometer that can be used with VKontakte data, namely, Akismet, Vkontakte Antispam and “Research weight of RuNet”. The first tool requires comment features that VKontakte does not provide, such as IP address. The second tool has not been updated for three years and is not working now. The third tool has also been removed from the Internet.

Another tool similar to Botometer exists for VKontakte. It is called GosVon and provides a database with VKontakte users labelled as bots[7]. However, the method on which the GosVon is based is unclear and not described on their website. There is no information regarding the performance and accuracy of this method. Moreover, the project aims to expose only pro-Russian bots, and it doesn’t take into account any other types of bots. Thus, the labels from this dataset are to be treated carefully and cannot be guaranteed to provide the ground truth.

Some of the user profiles on VKontakte contain a “deactivated” feature that can take the value of either “deleted” or “banned”. The banned users should most likely be bots. However, taking the banned users as a gold label for social bots is not the perfect approach to the model evaluation. The reasons for that are:

[7] https://gosvon.net/

- VKontakte moderators may ban users for different reasons than being a bot. For instance, real users spreading inappropriate content may be blocked.
- VKontakte moderators might miss some recently created bot accounts. Probably, they do not have enough time to check new bot accounts.
- VKontakte is a company with tight relations with the Russian government. Gazprom, a state-owned Russian gas company, has been the biggest shareholder of this social network since 2021[Tim]. Therefore, objectivity and transparency of VKontakte's moderators' work cannot be guaranteed.

Taking a look at the problem from a different perspective, we aim to understand which users are guaranteed to be real ones, not bots. The thesis author is registered on VKontakte since 2009 and has been an active user for more than 10 years. The thesis author is sure that all of her 204 friends on VKontakte are real people. Moreover, likely, the friends of the author's friends are also real people. Thus, if a user is a friend of the author or a friend of a friend, this user is probably not a bot.

In the collected dataset, there are only 3 author's friends and 308 friends of friends. These users can be seen as a standard of a real user with a high probability.

Another method by which it is possible to identify real users is by analysing the "verified" field returned from VKontakte. This field is true only for the users who have verified their identity using their official documents. Surprisingly, there are only 37 such users in our dataset of 5,6 million comments.

The two approaches above guarantee that 348 users out of 5,6 million are real humans. On its own, measuring how well the model performs on these users does not provide us with enough information to evaluate the model as a whole. However, evaluation on these users can be a part of the overall evaluation strategy.

The manual evaluation was also considered a possible evaluation method. However, this method presents several difficulties:

- The size of the dataset does not allow for a manual evaluation of the whole dataset in a reasonable amount of time.
- The accuracy of manual evaluation can be low. In one study, human evaluators have only been able to correctly label 24% of the bots[Cre+17]. "Humans can not detect sophisticated bots in most scenarios"[Kol+22].

In some papers regarding bot detection on VKontakte, e.g. [Kol+21] and [KCK21], the authors inject social bots to VKontakte and evaluate their models using these bots. While being a valid approach in other contexts, in the current research, it does not seem to be a suitable method. Injecting additional bots into VKontakte is not moral in the context of an ongoing armed conflict, can further destabilise the political situation in Russia and Ukraine and negatively influence the social landscape of the VKontakte community. Thus, we will not inject any bots spreading political propaganda in VKontakte.

As a summary of the above, the scientific community does not currently have a 100% reliable method of obtaining the ground truth for the bot detection task that this study aims to solve. Therefore, a combination of the most reliable methods will be used to evaluate the performance of models built in the course of this research. We will consider verified users

and those users who are in our friends network as real humans, and will additionally label a subset of users via crowdsourcing to gather independent opinions on whether some users from the dataset or bots or humans. This approach will allow evaluation of the model performance with some degree of certainty.

3.2.2 Labelling process

Due to the unreliability of individual judgment, we rely on several independent labels to obtain a summarized label for a small subset of users from the dataset. The labels are collected via a survey launched on the Prolific[8] platform. As a first step, it is necessary to find suitable respondents for the study. The respondents will have to inspect VKontakte accounts, posts and comments written by users from the dataset. Therefore, there are several criteria that the respondents should satisfy: they should have a VKontakte account and be fluent in both Russian and Ukrainian, since these two languages are the most popular in the dataset. Moreover, we filter the respondents by education level (at least Bachelor level) and approval rate (at least 95%). There were 250 such respondents on Prolific.

Next, we choose samples of users from the database to run the labelling process on. The first sample is entirely random and consists of 100 users. The randomness of the sample is ensured by using the $sample[9] aggregation stage in MongoDB. It is designated to be used to evaluate the Friendship relations method, described later in 3.2.4. The second and third samples consisted of 20 users each and are related to each of the methods used to build the bot detection model (URL sharing and Hashtag sequences). These methods are described in detail in further sections 3.2.6 and 3.2.7. The IDs of users to label were selected manually for the second and third sample from suspicious and normal clusters identified by the URL sharing and Hashtag sequences methods.

Labelling each user requires quite a thorough inspection of their profile and comments that they have written on VKontakte. We estimated that it would take each respondent up to five minutes to label each user. Therefore, the number of users to label per respondent was limited to 10 so as to take them less than an hour to complete. For each user from the samples, we aimed to collect at least three labels to obtain a more objective summarised label.

Launching the survey on Prolific required building a separate web interface to enable user labelling. This web interface is connected to the database via the web backend. To each respondent, this interface offers to firstly enter their Prolific ID in order to allow respondent identification later on. Next, the respondent is presented with a short description of the survey goals and the respondent's tasks. This description also contains the definitions of a social bot and real human account, so that respondents can better understand the context of the survey. The interface subsequently presents 10 users from the samples to the respondent and offers two buttons: "BOT" and "HUMAN". According to the personal judgement, after inspecting the user's VKontakte account and comments, the respondent decides which button to press. The user flow of the labelling process is shown in Figure 3.11.

[8] https://www.prolific.co/
[9] https://www.mongodb.com/docs/manual/reference/operator/aggregation/sample/

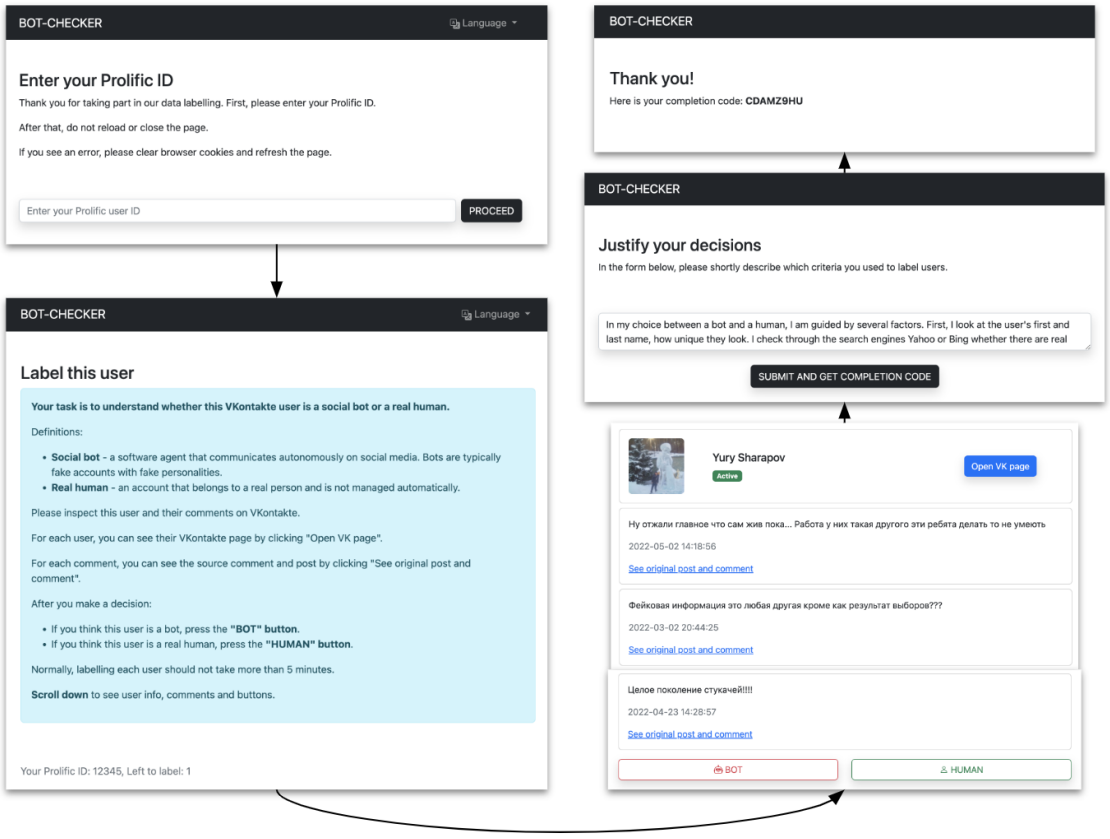

Figure 3.11: The labelling interface user flow

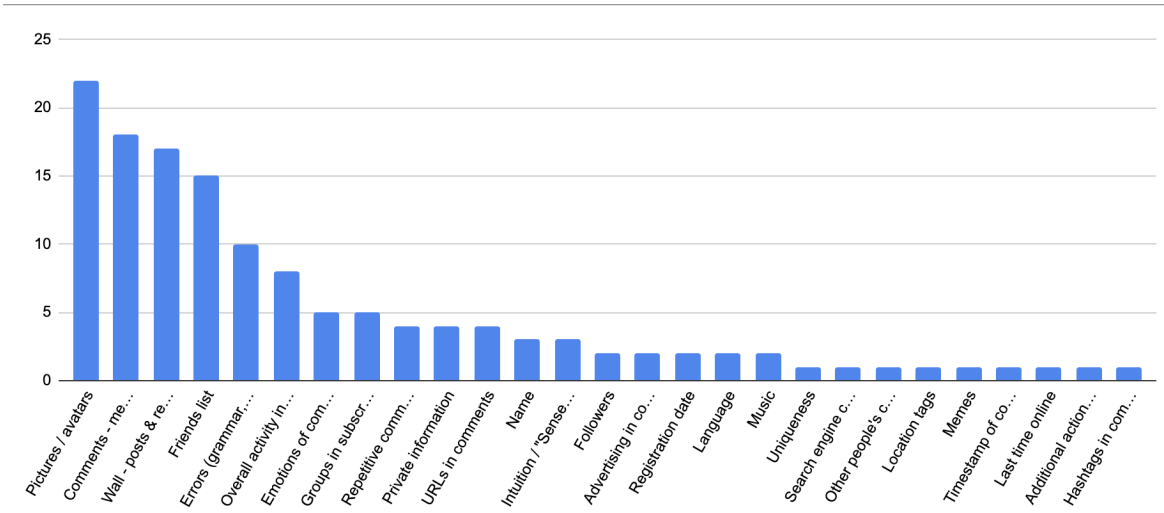

Figure 3.12: Characteristics that influenced labelling outcomes

In the end, respondents are asked to justify their labelling choices in a free-form text entry. By reading these comments, we can gain valuable insights into the features that respondents take into account when labelling users. From the respondents' free text responses, it was also obvious that one respondent was too subjective while labelling users. This person claimed that any VKontakte user with anti-Russian position was spreading fake news and propaganda by default and had to be labelled as bot. However, this user's labels did not skew the labelling results due to the presence of at least two other opinions for each user from the samples.

The characteristics that respondents used to label users are depicted in Figure 3.12. As we can see, most respondents paid attention to the photos, and especially to avatar pictures, of the users they checked. The following popular characteristic was the meaning of comments and the suitability of comments to the post. Respondents also paid attention to the posts and reposts of a user and to the friends list. Very few respondents checked for the features that might, according to the methods applied in this study, be indicative of bots. For example, only four respondents took into account the URLs or hashtags that VKontakte users share in their comments.

As a result of the labelling process, we obtained labels for 139 users from the dataset. One hundred users belonged to the first random sample. 20 and 20 users belonged to the second (URL sharing) and third (hashtag sequences) samples, respectively. There was an overlap in the case of one user who belonged to the second and third samples simultaneously. When calculating the resulting label, we adhered to the majority principle (e.g. if there were 2 "Bot" labels and 1 "Human" label, the resulting label was "Bot"). In case of a tie (13 cases), we labelled the user ourselves to obtain another opinion and calculate the resulting label. Out of 139 users labelled, 33 users were labelled as bots and 106 as humans. Therefore, the respondents considered 23% of the sample users to be bots. This percentage is higher than the number from previous research on VKontakte bots that claimed that 17% of users on this platform are bots[Cur22].

The resulting labels that we obtained as a result of the Prolific survey, along with other

account features, will be used as the gold standard to measure the performance of the model and draw conclusions about its efficiency.

3.2.3 Initial choice of method

The method selection phase consisted of the following steps:

a) Refining the search query. In order to avoid irrelevant methods, the search query should be as precise as possible. After several iterations of search query refinement, the most relevant results were obtained under the following query: unsupervised AND method AND "social network" AND "bot detection" AND(politics OR propaganda) -survey

b) Refining the time range. Since group-based approaches started gaining popularity only in 2018[Cre20], the time range was limited to 2018-2022.

c) Downloading the data. To do this, the scientific software Publish or Perish[10] was used. In total, 130 papers that matched the search query and the time range were downloaded.

d) Filtering out the articles that were never cited by any scientists. Since a higher citation score can indicate a more trustworthy paper, all the papers with a citation score of 0 were excluded. After that, only 81 articles remained.

e) Manually analysing the titles one by one. Titles summarize the papers and can give a general impression of the work described in each paper. During the manual selection, 53 papers were identified as potentially relevant to this thesis.

f) Preferably, the methods used in this research should have already been applied in the context of either politics in general or Russian politics in particular. Therefore, each article was identified as politics-related or not. In the politics-related ones, an additional classification for Russia-related and other papers was made. 14 of the papers were marked as politics-related, three of them as Russia-related.

g) Abstract analysis. The abstracts of the 14 papers were scrutinized to understand if these papers present methods applicable for the current research. Only 9 papers were left after this examination.

h) Paper analysis. The methods presented in the remaining 9 studies are further examined, and the following characteristics are identified (Table 3.1

- Approximate computational complexity and time. As the current research makes use of a large quantity of data, this characteristic is crucial for success. Lower computational complexity and time will allow running the method on all the data collected.
- Performance. This can be measured by F-score or ROC AUC in various papers. Bot detection quality is crucial to building an efficient and reliable bot detection model.
- Features and data used, and applicability to VKontakte. The method should be adaptable to VKontakte data specifics, such as the user or comment features, and should not depend on Twitter-specific features.

[10]https://publish-or-perish.en.softonic.com/

Table 3.1: Methods selected in the first iteration of model selection process

#	Type	Performance	Works for VK	Open code	Source
1	Supervised	Precision=78.5% ROC AUC=0,99	Yes	No	[Im+20]
2	Supervised	Accuracy=93,2-96,2% ROC AUC=0,98-0.99	Yes	Yes	[Pas+20]
3	Supervised	ROC AUC=0,79-0,89	Yes	No	[AO20]
4	Supervised	Accuracy=83%	Yes	Yes	[Ros+19]
5	Supervised	Accuracy=83%	Yes	No	[Ros+20]
6	Unsupervised	No data	Yes	No	[Niz+21]
7	Unsupervised	Accuracy=86%	Yes	Yes	[Hag+22]
8	Unsupervised	No data	Yes	No	[Han+19]
9	Unsupervised	Various	Yes	No	[Rio+18]

- Availability of the code. If the authors of a method provide open source code to support their work, it becomes easy to reproduce their research.

Judging by the Table 3.1, only 4 methods out of 9 are suitable for the current research due to their unsupervised method type. Out of these, only method 4 provides an estimate for accuracy. Moreover, method 7 provides the code to reproduce it. Therefore, for the current research, the seventh method is initially chosen.

The method was applied to the 2016 US elections data from Twitter. It consists of five consequent steps:

a) Employing a community detection algorithm to define the network structure and identify unique communities based on retweet relations.

b) Usage of a series of bot detection algorithms to identify the likelihood of each node in the network being a social bot.

c) Calculation of several commonly used centrality measures to identify influential actors in each of the detected communities.

d) Sentiment analysis to better understand the tone and content of the information communicated in the network as a whole as well as in each individual community.

e) Content analysis to describe the categories of user profiles in order to identify the types of actors who were most influential in the discussion network.

All of these steps can be applied to our dataset, except for the second. In the original paper, the authors use Botometer and tweetbotornot. There are no alternatives for these tools that could be used with VKontakte data. Therefore, this step is omitted. In step 1, the authors build a network of users based on the retweeting behaviour. Since there are no retweets on VKontakte, a different measure for building the network should be chosen. A graph of users can be built based on either of these aspects:

- Friends graph (as stated in [Kol+21], friends structure can be indicative of bots.

- Users similarity (building a weighted graph with user similarity as in [FA20]).

Since building the graph based on users' similarity would involve calculating the similarity metric more than 80 milliard times (283 506 squared), which is very computationally heavy and would take a significant number of days, a decision was made to base the graph on the friendship connections between users.

3.2.4 Friendship relations

In the original paper [Hag+22], the user graph was based on retweet behaviour. Since VKontakte does not have a tweeting/retweeting feature, an alternative should be found. As stated in [Kol+21], it is possible to identify bots by friend structure. Therefore, a graph can be built based on the friends' network. For each of the 283 506 users in the database, the additional field was parsed from VK API, containing the list of their friends. Then, for each user, this list is matched against the database users, and an intersection between the friends' list and the "users" collection is found. A graph depicting these friendship relations consists of nodes (represented by user IDs) and edges (representing the friendship relation between users).

To cluster the model, the Louvain clustering algorithm is used[Rit]. In the original paper[Hag+22], this algorithm is chosen because of the "easy implementation and high-quality results". In the end, the algorithm finds clusters with maximal modularity. Louvain clustering is widely used for community detection on social networks (for example, in [HKK16], [De +11], [Sán+16]).

For this study, the implementation of the Louvain algorithm from the popular and widely used python-louvain library[11] was chosen. This library takes advantage of NetworkX[12], a Python package for graph management.

The data about friends could only be fetched for 167 276 users. The other users have either hidden their friends lists from the public, or were deleted or banned. Therefore, only 167 thousand users have been used to form clusters. The user graph itself was formed by 73 253 nodes, each node corresponding to a user. These are the users that are connected to at least one other user in the graph. The Louvain clustering produced 5 478 unique clusters that were saved to a MongoDB collection.

After clusterisation, the Gephi software was used to create a visual representation of the clustered graph. Like in the original paper[Hag+22], we used the ForceAtlas 2 layout for the graph visualisation because it produced the best visually understandable cluster layout.

3.2.5 Subsequent choice of method

Building graph based on friendship relations only produces large clusters and might not be accurate. So in order to find a suitable method for more granular and precise bot detection, we now pay attention to the methods that were previously found during initial method selection but were filtered out some stage of the analysis. Instead of taking into account features like

[11] https://github.com/taynaud/python-louvain
[12] https://networkx.org/

open source code availability and applicability to VKontakte data, we rely more on qualitative analysis of titles, abstracts and full texts.

Out of 96 papers, we select 18 as the most suitable judging by the title and abstract. Having read through these 18 papers, we find one that presents a group-based unsupervised method applied to 5 different case studies with modifications. This is a study named "Uncovering Coordinated Networks on Social Media" [Pac+20]. The paper provides a core framework, on the basis of which five different models are built and applied to various datasets, including data from Hong Kong protests, US elections and other major political events. The framework is general and therefore versatile, allowing to hypothesise that it can be suited for the task of bot detection in VKontakte.

In the framework, several steps are proposed to detect bots:

a) Behavioral trace extraction. From a dataset containing tweets, we extract some behavioural features based on which users can be united into a network. These features, or traces, might indicate suspicious coordinated behaviour.

b) Bipartite network construction. A "user-to-feature" bipartite graph is constructed.

c) Projection onto account network. The bipartite graph is projected onto a user graph, and weights are derived by some transformation from the initial bipartite network weights.

d) Cluster analysis. The resulting user network is analysed manually to identify "suspicious" and "normal" clusters with high potential number of bots or humans, respectively.

To this framework, we add an additional step: comparison of the model results with the labelling results (3.2.2). This allows to estimate each model's performance using standard metrics, such as accuracy, precision and recall. We then compare the resulting models and make conclusions about each model's applicability to the case of bot detection on VKontakte.

The success of a bot detection model built with this framework depends on the choice of initial behavioural traces considered suspicious. In the five case studies, these traces are different:

a) Account handle sharing;

b) Image coordination;

c) Hashtag sequences;

d) Co-Retweets;

e) Synchronised action.

From these five types of traces, only Image coordination, Hashtag sequences and Synchronised action apply to our dataset. Moreover, the Synchronised action method is the least efficient and therefore not suitable for the large dataset size. Thus, in the subsequent sections, we will explore the application of Image coordination and Hashtag sequences methods to the task of this research.

3.2.6 Image and URL sharing

The method used in [Pac+20] to detect coordinated communities of bots during the Hong Kong protests of 2019 utilises image similarity to construct a user graph. More specifically,

it first examines tweets by various users, extracts the images from them and creates RGB histograms for each image. At first, the same method was meant to be applied to our dataset. However, we soon realised that VKontakte API does not allow obtaining raw image data by image URL. Therefore, when parsing image URLs from comments, we could not create histograms for VKontakte images. It was only possible to do so for images hosted on some external websites. However, the majority of users shared only VKontakte images. Therefore, building a complete graph with image histograms was not possible.

Instead of building a graph based on image similarity, we have built it based on URL-sharing patterns. The approach was similar to the image similarity method, except that for links, there was no need to create any histograms. They could just be compared for equality. The complete URL sharing method consisted of the following steps:

a) Retrieve URLs from all comments using a regular expression;

b) Create a bipartite graph that connects users to URLs, where edge weights are determined by how often a user shares a specific URL;

c) Perform a projection of the bipartite network to obtain a weighted account coordination network, with weights of edges calculated using the bipartite network weights with Jaccard similarity metric;

d) Visualise the graph with Gephi and identify "suspicious" and "normal" clusters.

3.2.7 Hashtag sequences

The Hashtag-sequences method from [Pac+20] is also applied to the same dataset in order to build a different group-based model and compare it with the URL-sharing model. The Hashtag-sequences model is based on the same steps as the URL-sharing model. However, the main feature with which we capture account similarity is not the similarity of URL that users share, but the similarity of hashtag sequences they share in their comments. In the original study[Pac+20], this method is applied to the case of US elections discussion on Twitter.

3.2.8 Evaluation

To evaluate and compare the three bot detection methods, we calculate standard metrics:

- Accuracy: $(TP + TN)/(TP + FN + TN + FP)$
- Precision: $TP/(TP + FP)$
- Recall: $TP/(TP + FN)$

Accuracy tells how many times the model was correct overall. Precision is how good the model is at predicting a specific category. Recall tells how many times the model detected a specific category[13]. The higher all three metrics, the more successful we consider a bot detection model. It is not sufficient to have just one or two of these metrics with a high value because that might mean that a model is good at predicting one class (e.g. humans) but fails at predicting the other (e.g. bots).

[13]https://www.mage.ai/blog/definitive-guide-to-accuracy-precision-recall-for-product-developers

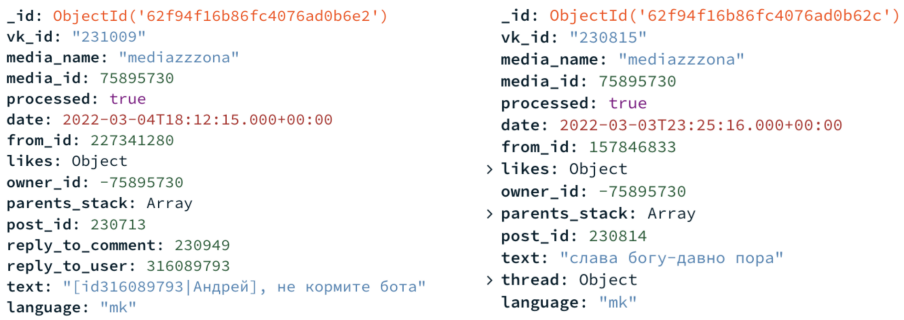

Figure 3.13: Comments containing Russian text and misclassified as Macedonian

3.2.9 Exploring influentialness

To identify influential users in clusters, a centrality metric Degree Centrality and a clustering coefficient are calculated. To compute these values, standard NetworkX functions are used.

Degree centrality can be a measure of influentialness of an actor in the graph[Hag+22]. The clustering coefficient "of node A measures the extent to which the neighboring nodes of A form a densely clustered clique"[Hag+22]. Higher clustering coefficients in a network show stronger connections among actors in that community[Hag+22].

To estimate bots' influentialness, we calculate these two metrics for two user networks: before bot removal and after bot removal. This allows us to see how the influentialness metrics change due to bots' presence. The same "before-and-after" approach is applied to sentiment analysis.

3.2.10 Comment language detection

Since the dataset may contain content produced in various languages, including, primarily, Russian and Ukrainian, but also potentially English, it is crucial to identify the language in which each comment is written. This will help in further sentiment analysis. To detect language, a popular langdetect[14] Python library is used. It is a re-implementation of Google's Java language-detection library.

Surprisingly, after the first iteration of language detection for the comments in the database, the third and fourth most popular identified languages were Macedonian and Bulgarian.

However, after a close manual examination of a sample of 1000 comments, it became clear that comments labeled as "Macedonian" contained of Russian text in 99% of the cases. Examples of comments misclassified by the langdetect library are given in Figure 3.13.

The same could be observed for comments identified as containing Bulgarian language. Therefore, all the comments classified with "mk" or "bg" language codes were updated to be classified as Russian text.

After that, the distribution of languages became as displayed in Figure 3.14.

[14]https://pypi.org/project/langdetect/

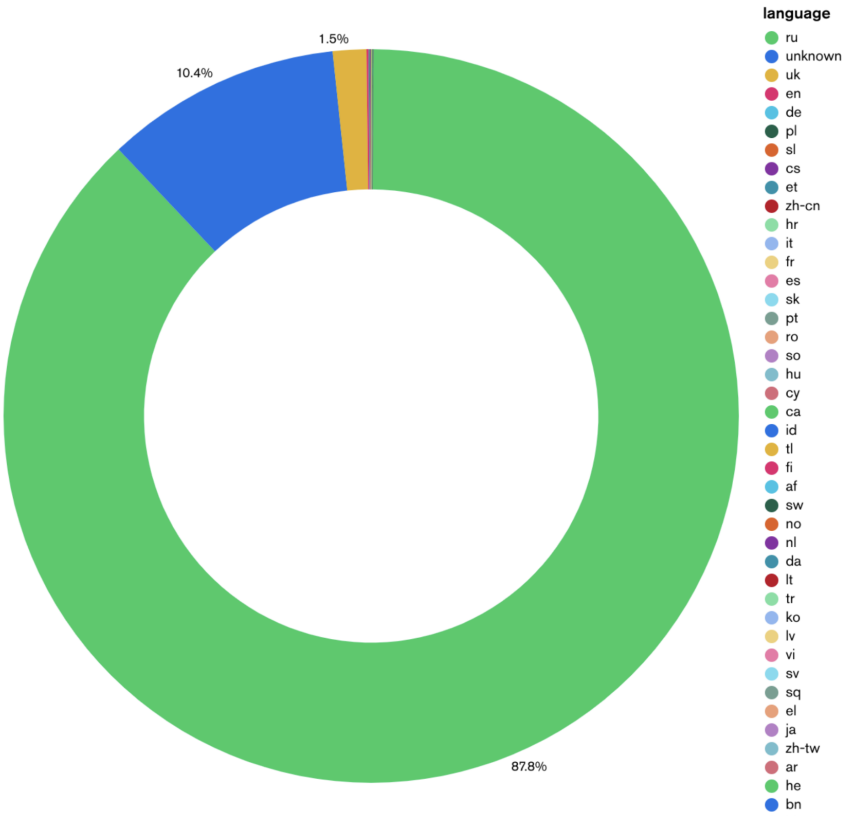

Figure 3.14: Distribution of comments by language

The most popular language in the dataset is Russian with 87.8% of comments. The second most frequent value is "unknown" (10.4%). This label is used in four possible cases:

a) When a comment consists of user tagging without any additional text;

b) When a comment consists of emoji only;

c) When a comment does not contain any text;

d) When the langdetect library cannot detect a language for the comment.

The Ukrainian language is the third most frequent value in our dataset with 1.5% comments written in this language.

The rest 0.3% of the comments were identified as written in languages other than Russian and Ukrainian. After a manual examination, we came to the conclusion that the language was identified incorrectly for most of these comments. In absolute numbers, the number of incorrectly identified comments is quite big (about 17 thousand comments), but relatively, it's a small fraction of the dataset. Therefore, for further analysis, these comments will not be taken into account for the sake of simplicity.

As a result of the language detection step, the "language" attribute was defined for each valid comment in our database. The most popular languages are, as expected, Russian and Ukrainian. Russian language dominates over Ukrainian in our database. This can be explained by the fact that most of VKontakte users (82.38% come from Russia, while only 3.22% come from Ukraine[Sim22].

3.2.11 Exploring comment sentiment

Conducting sentiment analysis on comments written in English or Russian is quite a widespread task that has established popular solutions. In the original paper where the authors analysed English content, they used the SentiStregnth model. It currently supports 16 languages. However, Ukrainian is not one of them. Moreover, the Ukrainian sentiment analysis research does not offer widely adopted methods and models. Even in the most recent articles dedicated to the Russian-Ukrainian conflict of 2022, sentiment analysis is conducted on texts written in English (e.g. [CSD22]). Thus, conducting direct sentiment analysis for Ukrainian texts presents an unsolved task and cannot be applied to this study. Instead, as Russian and Ukrainian share a common Slavic root, we hypothesise that translating Ukrainian comments to Russian will not lead to a significant loss in meaning. After this translation, it will be possible to conduct sentiment analysis on the translated text.

Just as in the original paper[Hag+22], we use the SentiStregnth model. The Python version of this library[15] allows integrating the sentiment analysis into our data processing pipeline. SentiStrength outputs scores from 1 (not positive) to 5 (extremely positive) for the "positivity" of a piece of text and scores from -1 (not negative) to -5 (extremely negative) for the "negativity" of text. Both scores were calculated and saved to the database for all the comments written in Russian and Ukrainian language.

[15] https://github.com/zhunhung/Python-SentiStrength

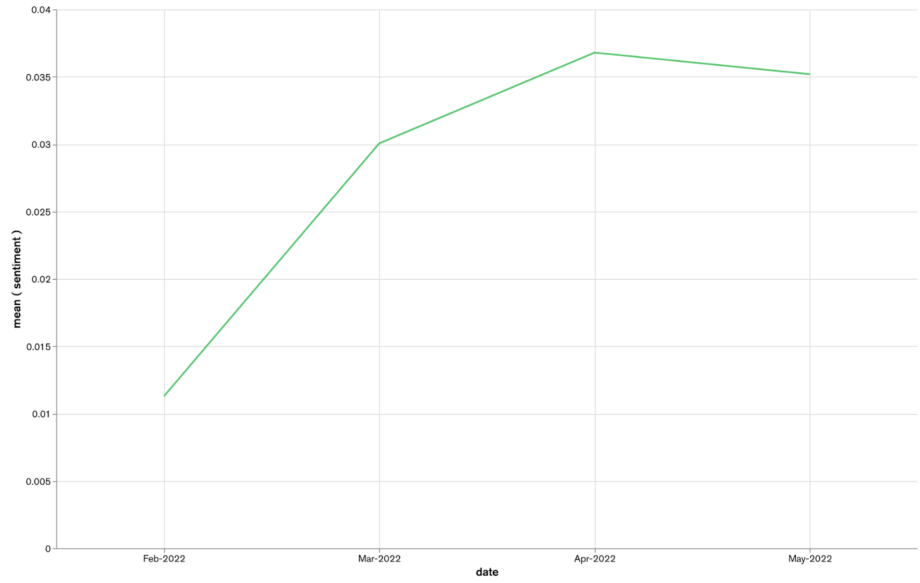

Figure 3.15: Average comment sentiment over months

As can be seen from Figure 3.15, the average comment sentiment is slightly positive but varies over time. The most negative sentiments are observed in February 2022. Potentially, this can be explained by the fact that the escalation of the armed conflict just started in February, and people were confused, stressed and did not know what to expect. Then, the average sentiment goes up in March, and even higher in April. Still, the absolute sentiment values are small and far from being "definitely positive". The sentiments are just slightly positive. In May, the sentiment becomes a bit more negative. The fluctuations in the level of sentiment can be explained by the events that occurred in the corresponding months. However, they can also be based on the presence of bots in the online environment.

3.3 Development of the web tool

3.3.1 Purpose

A bot-checking tool developed as a result of this research should present a web interface available to the public. In this tool, a user should have the possibility to enter the ID or name of a VKontakte user. Then, with the use of the combined model developed in the course of this research, the system should identify whether a user with this ID or name is a bot or not and shows the result to the requesting user. The closest existing analogue is Botometer[16]. However, the difference is that our bot checking tool should work without authentication from the user side and analyse VKontakte accounts instead of Twitter ones.

[16]https://botometer.osome.iu.edu/

3.3.2 Requirements

Based on the purpose and the core functionality of the web tool, the following requirements were identified:

a) Functional:
   a) A user must be able to search for VKontakte users from our dataset using either of these values:
      i. VKontakte ID;
      ii. First name;
      iii. Last name.
   b) A user must be able to see a list of users matching the search query.
   c) A user must be able to check whether a VKontakte user is a bot or a real human, according to the clustering done by our bot detection model.
   d) A user must be able to see the description of the method used to identify bots so that transparency of the system is ensured.
   e) A user must be able to see the web interface author's contacts so they can get in touch with the author if they have questions or comments.

b) Non-functional:
   a) The system should be available without registration or authentication.
   b) The system should be publicly available on the Internet.
   c) The system's design should be accessible.
   d) The system's design should be responsive to different screen sizes.
   e) The system should be available in three languages:
      i. English
      ii. Russian
      iii. Ukrainian

The requirements provided in the list above are essential for the development of the bot detector web tool. Their implementation is described in the next section.

3.3.3 Implementation of the web interface

The web interface was implemented based on the Flask[17] framework for Python. Flask allows building web applications in a minimalistic and easy way, providing features both for backend and frontend development with Jinja templates.

In order to ensure the responsiveness and accessibility of the system, an open-source CSS framework Bootstrap[18] was used. It provides design templates for typography, forms, buttons and other web UI elements.

[17] https://flask.palletsprojects.com/en/2.2.x/
[18] https://getbootstrap.com/

3.4 Threats to validity

Sentiment analysis was an essential step in this study. However, the sentiment analysis might have yielded imprecise results for 1,5% of the dataset (comments written in the Ukrainian language). Since there is no reliable and widely accepted model for sentiment analysis of Ukrainian texts, we analysed the translated version of the comments. During the translation, some parts of the meaning could have been lost.

The absence of ground truth presented a major obstacle and limitation to this research. In general, this is an almost unsolved problem in the whole modern bot detection research field. The most reliable way to obtain the ground truth is to create and inject social bots into social networks. However, this technique did not seem ethical in the context of the ongoing armed conflict. Other tactics of getting the ground truth currently seem to be approximate. Therefore, it is challenging to evaluate the quality of the bot detection models reliably.

The URL-sharing model, although efficient in detecting specific bots, can only detect coordinated communities that share URLs as part of their influence methods. Other types of bots are not detected by this model.

The web bot detection tool developed during this research only allows retrieving the data about users from the collected database and not about any VKontakte user. Therefore, the applicability of the tool is limited.

4 Results

4.1 Automatic bot detection

4.1.1 Friends graph

In Figure 4.1, you can see the visual representation of the graph produced with ForceAtlas 2 layout algorithm. In various bright colours, the largest clusters (clusters that contain more than 2% of the users in the dataset) are shown. The remaining clusters are coloured grey.

Even though the clusters can be visually identified, they are quite tightly connected. In Figure 4.1 we can notice at least seven major clusters: clusters number 7 (lilac), 10 (green), 21 (orange-red), 8 (yellow), 2 (crimson), 16 (lighter blue), 15 (aquamarine). These clusters present a great interest for further investigation because they are quite large compared to the rest of the clusters and are the most visually separable from the rest. The clusters 25 (brown), 1 (dark blue) and 3 (pink) are also visible but are not so clearly separable from one another.

Next, we explore the distribution of users that we definitely know or suspect to be real humans, throughout the graph. These users include verified users and "friends and friends of friends" users.

For the verified users, it is known for sure that they are real humans: they undergo a document check before receiving the "verified" label. The placement of verified users on the user graph is shown in Figure 4.2 in bright green colour.

It can be clearly seen that the position of all verified users is skewed to the left, with very few of them on the right side of the graph. Thus, the concentration of users whom we definitely know to be human is zero in almost all noticeable clusters that are separable from the other ones. Taking a look at the major clusters mentioned above, we notice that in clusters 7, 15 and 3 there are no verified users. Clusters 1 and 2 contain only one verified user each. At the same time, the major clusters 10, 21, 25 contain much more verified users.

Next, the concentration of "friends and friends of friends" of the thesis' author is explored. In Figure 4.2, we can also see a skew to the left, with very few users on the right side of the graph. A lot of such users belong to the major clusters mentioned above, e.g. 6 users belong to cluster 1, 29 users to cluster 2. Cluster 15 does not contain any such users.

It is crucial not only to identify the position of potential real humans on the graph, but that of bots, too. The only probable indicator that we have for bots is the "banned" status from VKontakte API. The banned users are displayed as right dots in Figure 4.2.

It is possible to divide all the clusters into three categories:

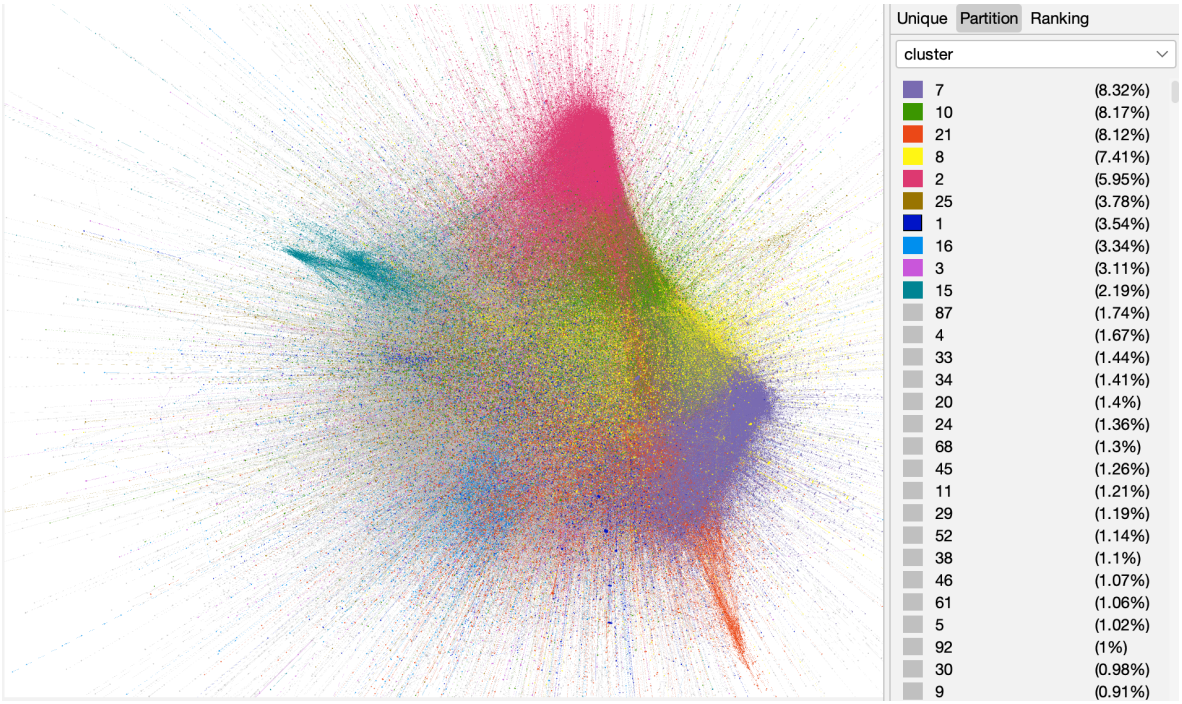

Figure 4.1: User graph clustered with Louvain algorithm, ForceAtlas 2 layout

- Clusters that contain a high number of banned users and low number of verified and "friends of friends" users. These clusters are highly likely to consist of bots.
- Clusters that contain a low number of banned users and high number of verified and "friends of friends" users. These clusters are highly likely to consist of real human users.
- Clusters that either do not contain verified, friends and banned users, or contain them in an equal proportion. These clusters are unidentified by the model, by default we assume they are human.

To identify the thresholds applied to the number of banned and verified users in a cluster, we adhere to a rule that the number of bots in our network should be around 1%, as previous research shows that is the typical number for VKontakte[Cur22].

The logic of applying the thresholds to the clusters was the following:

- If the percentage of banned users in a cluster is higher than the 'banned_threshold' that equals 0;
- And the percentage of verified users in a cluster is lower or equal than the 'verified_threshold' that equals 0.0005;
- And the percentage of "friends and friends of friends" users is lower or equal than the 'friends_threshold' that equals 0.0025;
- Then the cluster is identified as a potential bot cluster. Otherwise, it is a human cluster by default.

By choosing the thresholds for the level of banned, verified and friends users, the following results are achieved. 6 clusters are identified as potential bot clusters (clusters number 1, 3,

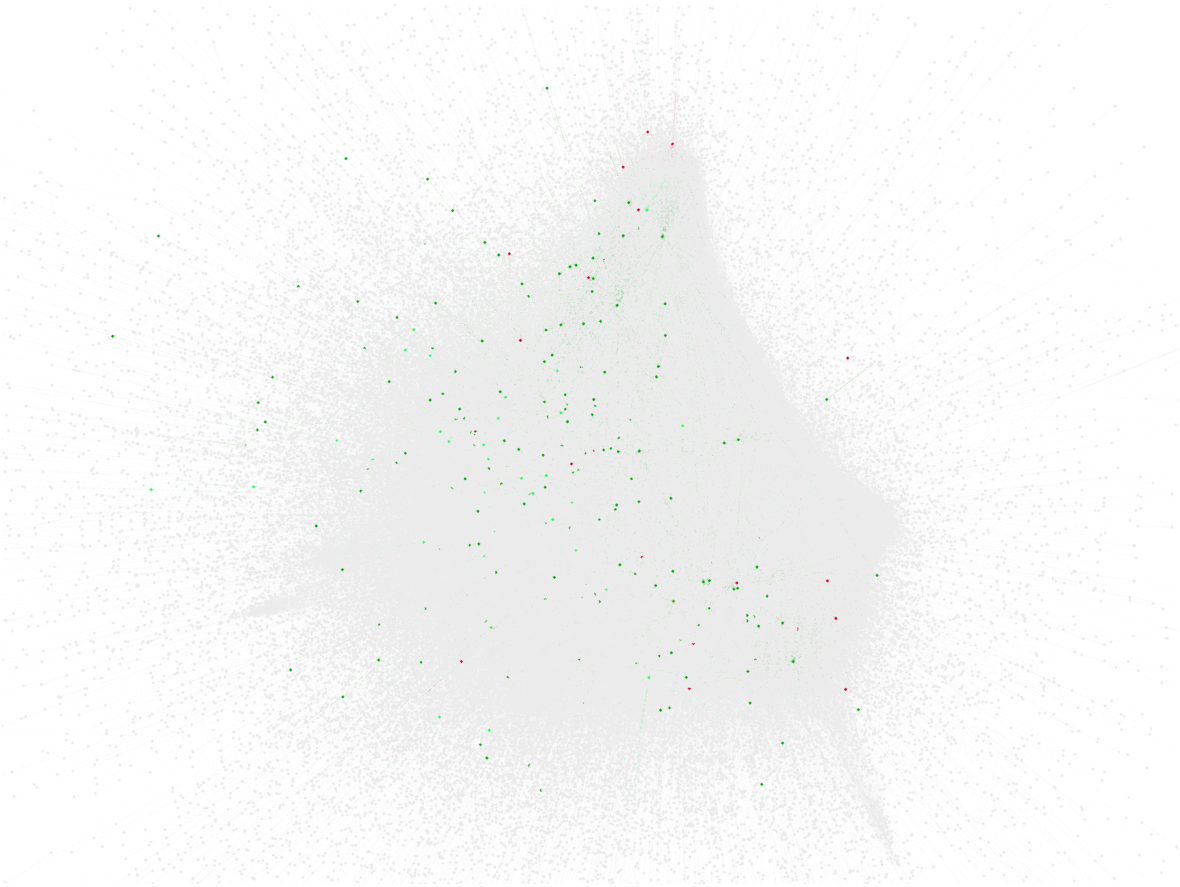

Figure 4.2: User graph with verified (bright green), banned (red) and friends (light green) users

7, 24, 35, 158), or "suspicious" clusters. These 6 clusters contain 16,3% of the total number of users. These 16,3% of users have left 22,3% of the total comments saved in the database.

We next compare the predictions of the model to the results of labelling and compute the key metrics for the model. The values of these metrics are the following:

- Accuracy: 64.29%;
- Precision: 25.00%;
- Recall: 33.33%.

As we can see, the performance characteristics of this model are unsatisfactory. Changing the threshold values did not improve the metrics. We next hypothesise that the clusters in the user network built on friendship relations are too big and do not allow for the precise identification of bots. In order to build more granular clusters, a different method is needed that allows uncovering hidden connections between users that are not as obvious and widespread as friendship relations.

4.1.2 URL sharing

The final user network contains 986 nodes connected with 1668 edges. Figure 4.3 displays the user graph that was obtained with this method. Each node is coloured according to the node's degree. The higher the degree, the darker is the shade of green. Moreover, the clusters in this figure are labelled with identifiers U1-U10. Clusters U1-U6 are "suspicious" clusters where the model predicts that the users are bots. The suspicious clusters are easily identified as separate groups of users, usually tightly coupled nodes with a high node degree. Clusters U7-U10 are "normal" clusters where the users are supposedly humans.

The URL sharing model is evaluated using the labels that were collected in 3.2.2. Standard key metrics are calculated to measure model performance:

- Accuracy: 85.00%;
- Precision: 83.33%;
- Recall: 90.91%.

Judging by the metrics, the URL sharing model is capable of efficiently identifying bot clusters and human clusters. Dense clusters typically consisting of 5-20 users where users frequently share the same URLs are highly indicative of bots. These clusters are easily separable from the rest of the nodes in the graph and have a distinct structure where either one account is symmetrically connected to all the other accounts or all accounts are tightly coupled between each other. Small clusters with low node degrees and asymmetric clusters, on the opposite, indicate the presence of real human accounts.

The users from suspicious bot clusters share the same links, the most popular of them are the following:

- Cluster U1: Telegram channel, VKontakte and Odnoklassniki group "dnevniksfronta" (pro-Russian); negative news about the USA; YouTube video of an explosion in Ukraine; Telegram channel "redakciya_channel" (pro-Russian).

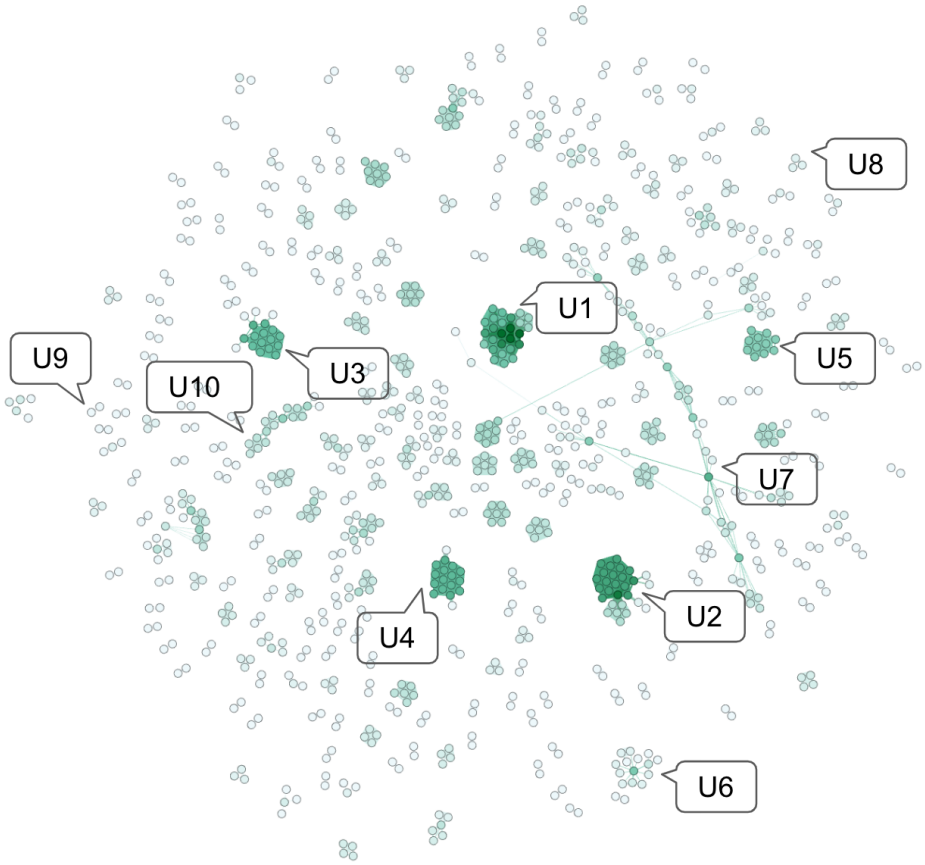

Figure 4.3: User network built with URL sharing method

- Cluster U2: the website 200rf.com (a website launched by Ukraine to help Russian families find their relatives that were killed during the conflict; currently not available on the Internet)[1].
- Cluster U3: the Wikipedia article about Russian invasion of Ukraine[2]; anti-war YouTube interviews.
- Cluster U4: the website 200rf.com, YouTube videos.
- Cluster U5: Telegram channel "warfakes" (pro-Russian).
- Cluster U6: pro-Ukraine Telegram channels and YouTube videos.

As we can see, these users most frequently share Telegram channels and YouTube videos, as well as external websites and VKontakte groups. For each cluster, it is possible to identify the attitude towards the Russian-Ukrainian armed conflict and the purpose of URL sharing.

It is also interesting to find out whether the bots that the URL sharing method has found were detected by VKontakte moderators and banned. To do so, we visualise the user statuses on the user network, showing banned users in bright pink (Figure 4.4).

As we can see, some bot clusters contain a high number of banned users (e.g. cluster U1 and U6). This means that users from these clusters were likely banned because VKontakte moderators considered them bots. On the other hand, these clusters contain a few active users that are not banned yet. Moreover, bot clusters U2-U4 contain very few banned users and bot cluster U5 contains none. This means that these bots have escaped being banned on the platform.

As a result of applying the URL-sharing method, a model able to successfully identify bot and human users was created. The model has high accuracy, precision and recall values and identifies bots that avoided banning on VKontakte. URL-sharing group-based method is both an easy and efficient method of detecting coordinated groups of bots in the discourse about the Russian-Ukrainian armed conflict.

4.1.3 Hashtag sequences

As a result of applying this method, we obtain a user network consisting of 384 nodes connected with 4087 edges. While this graph contains three times less nodes than the URL-sharing graph, the nodes are a lot more connected and form tighter clusters. This can be seen in Figure 4.5.

The suspicious clusters seem to be easy to identify. We consider clusters H1-H5 suspicious and clusters H6-H7 normal. This aligns with the principle based on which we split URL-sharing clusters into suspicious and normal. The tighter a cluster is and the higher node degree is, the more suspicious does the cluster look.

Surprisingly, the Hashtag-sequences method does not show good results when evaluated on summarised labels. After calculating the key metrics, we have obtained the following results:

[1] https://www.pravda.com.ua/eng/news/2022/02/27/7326424/
[2] https://en.wikipedia.org/wiki/2022_Russian_invasion_of_Ukraine

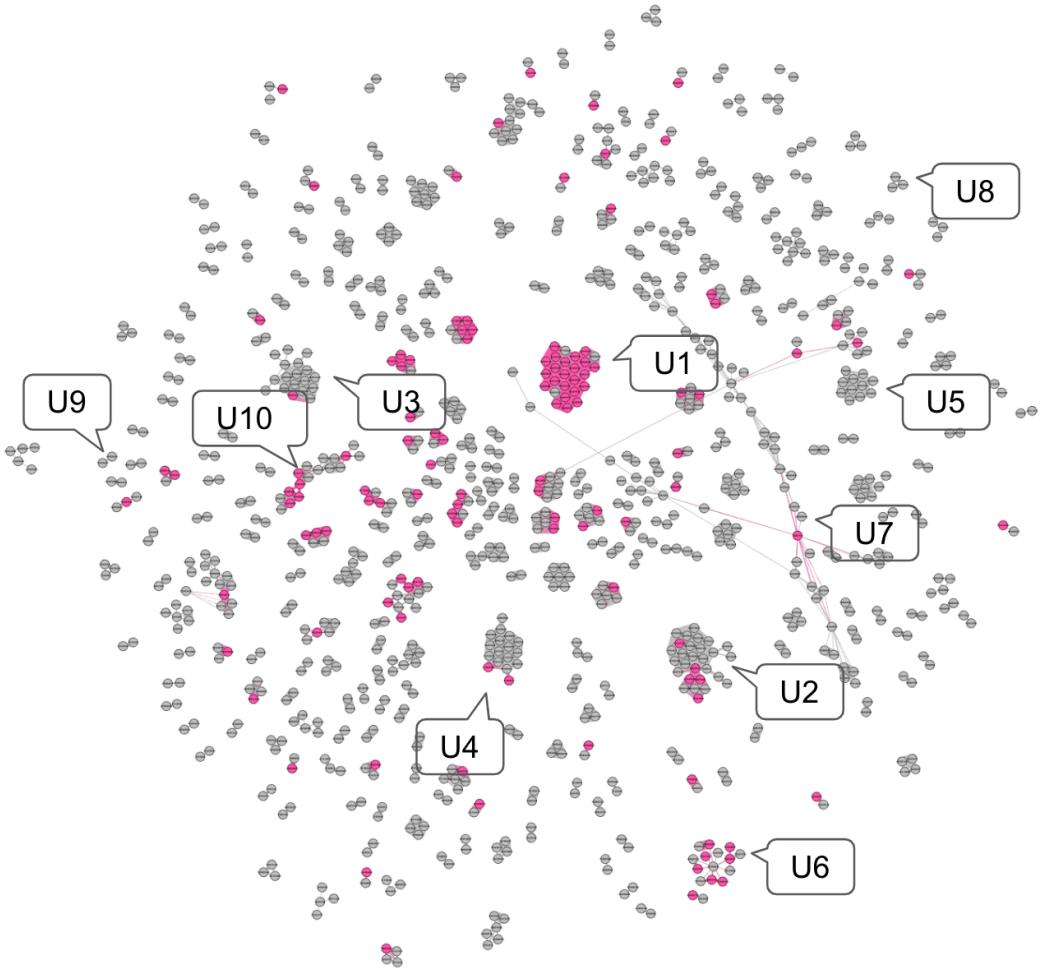

Figure 4.4: User network built with URL sharing method, with banned users in pink colour

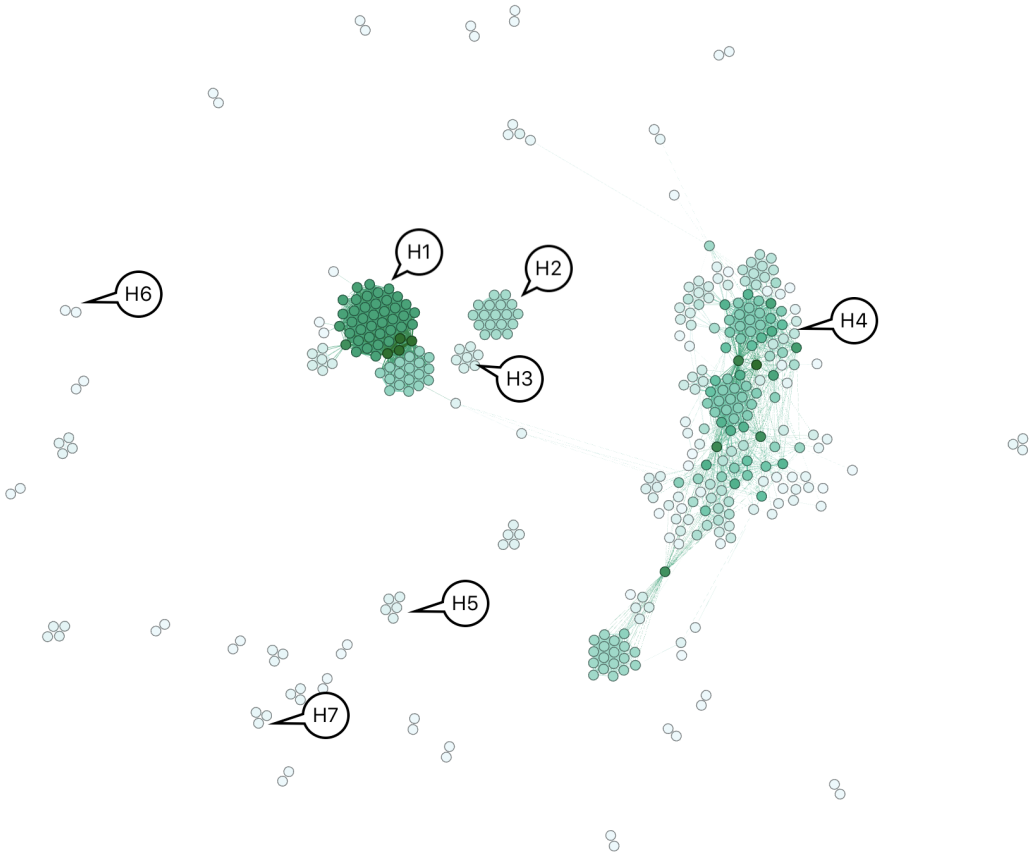

Figure 4.5: User network built with Hashtag-sequences method

Table 4.1: Comparison of main statistical measures of centrality metrics before and after bot removal

Metric	Statistic measure	Full network	Filtered network	Difference
Degree centrality	Mean	0.0034	0.0020	0.0014
	Standard deviation	0.0040	0.0013	0.0027
	Maximal value	0.0254	0.0051	0.0103
Clustering coefficient	Mean	0.4851	0.4084	0.0767
	Standard deviation	0.4848	0.4832	0.0016
	Maximal value	1.0000	1.0000	0.0000

- Accuracy: 33.33%;
- Precision: 17.65%;
- Recall: 100.00%.

Only the recall metric is high for this method, and it's because the number of False Negative values for this method is zero. The problem of the Hashtag-sequences method is that it labels users as bots way more frequently than needed. Our human respondents labelled most of the "suspicious" users as real humans. Therefore, we cannot consider this method reliable and efficient for the task of bot detection in the discourse about the Russian-Ukrainian armed conflict.

4.2 Bot influence in the user network

4.2.1 Influentialness

As can be seen from Table 4.1, after the removal of potential bot clusters, the centrality metrics distributions have changed significantly. Average degree centrality sinks. Clustering coefficient becomes significantly lower than before the bot removal.

The decrease in centrality metrics and influentialness of the network aligns with the findings of the paper[Hag+22]; the same behaviour was observed after the removal of bot accounts.

4.2.2 Sentiments

After the potential bots removal, the range of positive, negative and overall sentiment scores becomes more narrow. This can be observed in Table 4.2.

The findings of the sentiment analysis suggest that bots influence the sentiments in the user network, primarily by amplifying negative sentiments. They also slightly influence the positive sentiments. This might be explained by the fact that bot creators aim to provoke emotions and emotional decisions among the people reading bot-produced content, to replace the rational thinking with emotions.

Table 4.2: Comparison of main statistical measures of sentiment scores before and after bot removal

Metric	Statistic measure	Full network	Filtered network	Difference
Positive sentiment	Mean	1.2164	1.2138	0.0026
	Standard deviation	0.4662	0.4730	-0.0068
	Maximal value	3.0000	3.0000	0.0000
Negative sentiment	Mean	-1.1973	-1.1775	0.0198
	Standard deviation	0.5097	0.5053	0.0044
	Maximal value	0.0000	0.0000	0.0000
Overall sentiment	Mean	0.0096	0.0182	-0.0234
	Standard deviation	0.1963	0.1960	0.0003
	Maximal value	1.0000	1.0000	0.0000

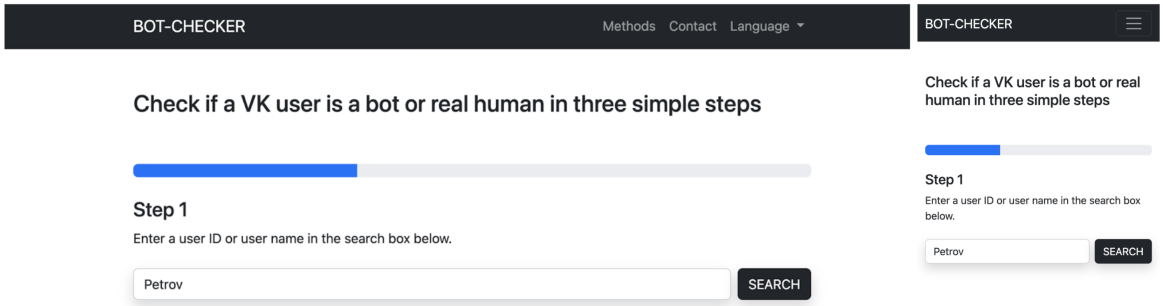

Figure 4.6: Search screen of the web application, desktop and mobile version

4.3 Availability of bot detector to the public

The screens of the web interface are displayed in Figures 4.6, 4.7 and 4.8

Figure 4.6 depicts the first step of the user's journey. This step allows user to type a user ID or first/last name in order to launch a search for a VKontakte user in the database. After pressing the "Search" button, a user is redirected to the search results screen (Figure 4.7).

In Figure 4.7, the search results of the query are displayed. The next step for the user is to choose the matching VKontakte user and press "Check" to learn if this VKontakte user is a bot, according to our model's predictions, or not.

The last step of the user journey is shown in Figure 4.8 Here, the results of the VKontakte user check are displayed.

The web application presents two other pages: Methods and Contact (Figures 4.9 and 4.10, respectively).

The page displayed in Figure 4.9 gives a top-level overview of the method used for bot detection. Using this page, the user of our system can get acquainted with the technology

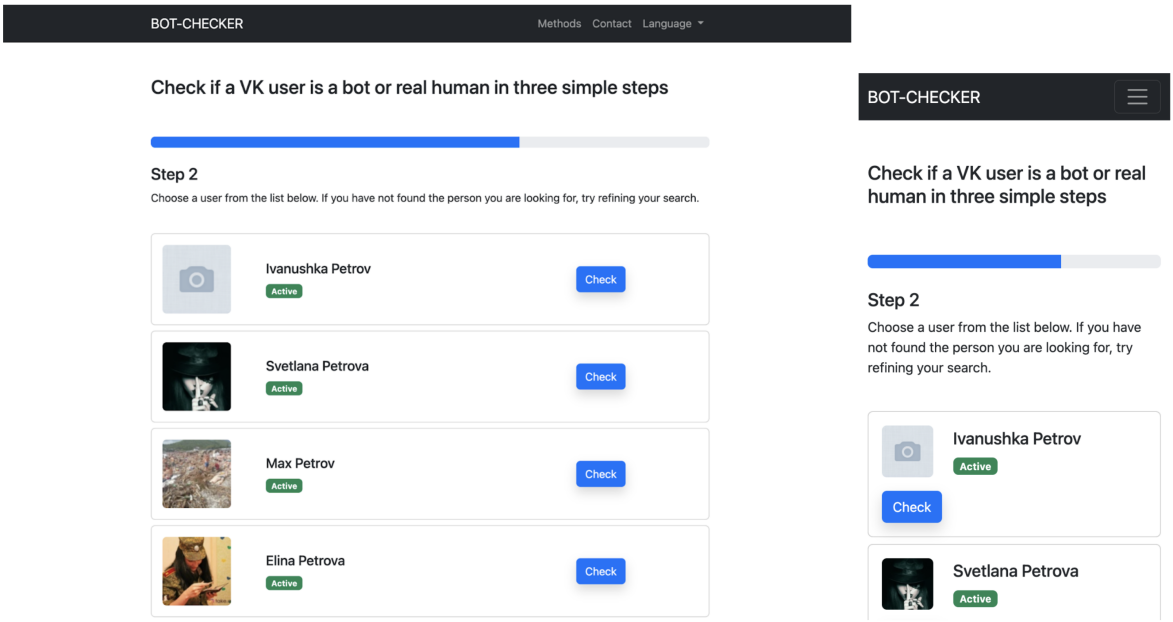

Figure 4.7: Search results screen of the web application, desktop and mobile version

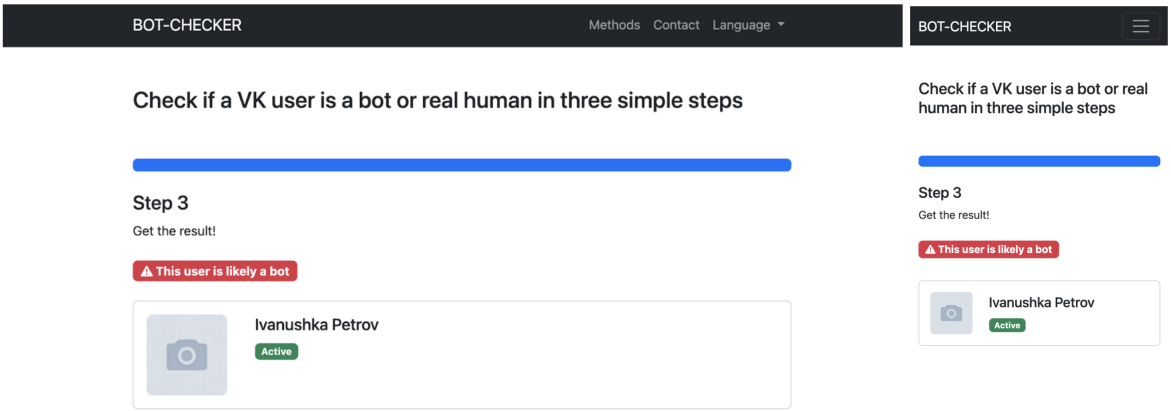

Figure 4.8: User check result screen, desktop and mobile version

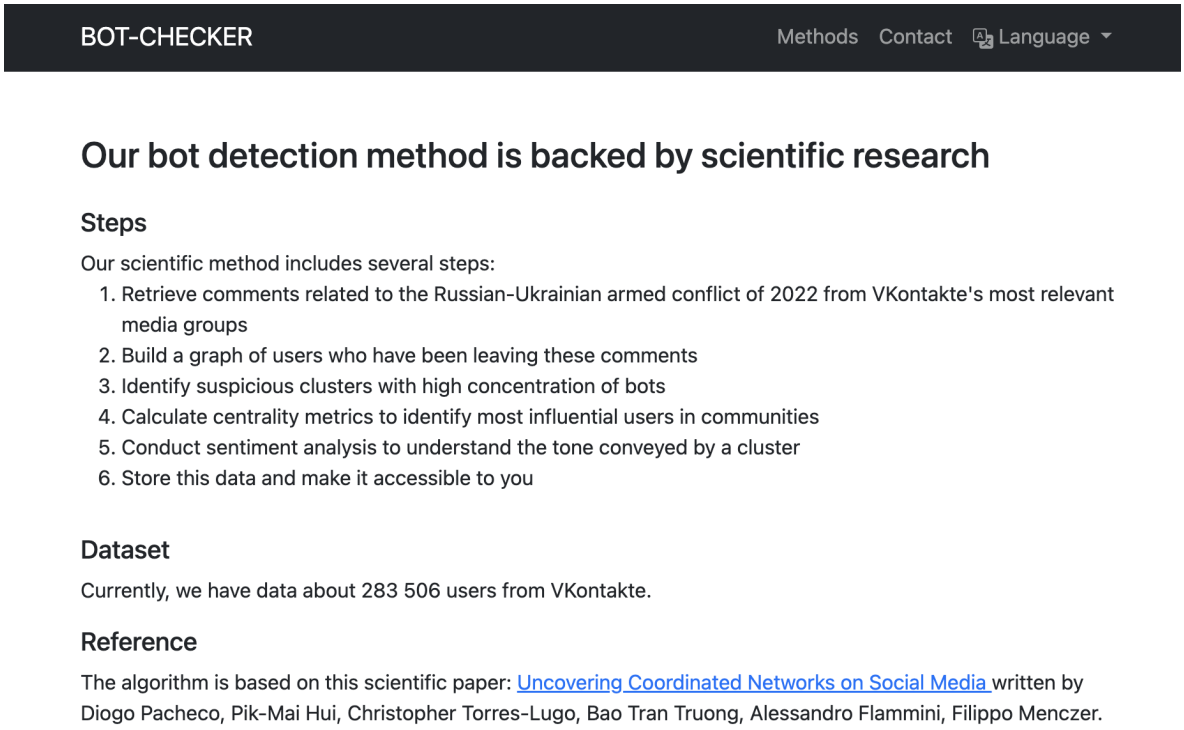

Figure 4.9: Methods screen, desktop version

that lies at the core of the system and sees that the process of bot detection is transparent and based on scientific research.

If a user has any questions, they can use the Contact page depicted in Figure 4.10. Here, they can get in touch with the author of this work and ask a question or report a problem. The Methods and Contact page are also responsive to various screen sizes.

Since the system should be multilingual, a language switch control has been implemented. It is available in the header of the website throughout the user journey. The web application is localised with the usage of Babel library[3]. An example of the localised interface is given in Figure 4.11

After the implementation of the web tool, it is necessary to deploy it to a remote server in order to make it accessible to a broad audience. The web application was deployed to a DigitalOcean[4] server. As of March 2023, the system is publicly accessible at the URL address `https://bot-detector-x2wo4.ondigitalocean.app/`. The complete source code is available at `https://github.com/lerastromtsova/malicious-bot-detection`.

During the web interface implementation and deployment, a system satisfying the functional and non-functional requirements from section 3.3.2 was built. As a result, any user who has

[3]https://babel.pocoo.org/en/latest/
[4]https://www.digitalocean.com/

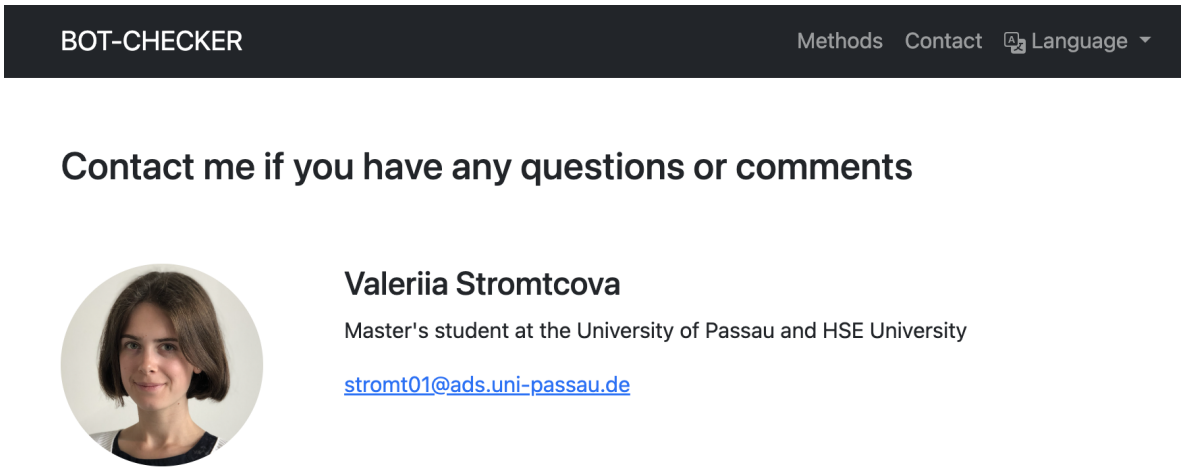

Figure 4.10: Contact page, desktop version

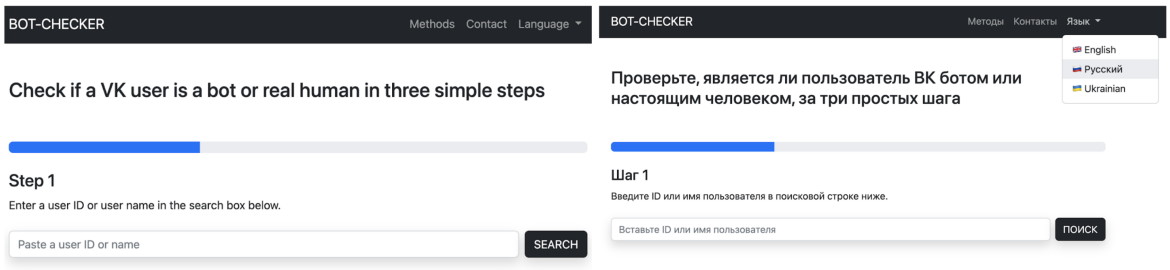

Figure 4.11: Localisation of the search screen, English and Russian versions

access to the Internet can check if a given VKontakte user is, according to the predictions of our model, a bot or a real person. This can help OSN moderators to more efficiently identify bots, and ordinary users to understand when their opinion is probably being manipulated and uncover this manipulation.

5 Discussion

As a result of this research, firstly, a dataset of VKontakte discussion around the Russian-Ukrainian armed conflict in 2022 was collected and analysed. Then, three different group-based bot detection models were built and applied to this dataset. The first model is based on friendship relations between users and allows creating of a graph spanning more than 73 thousand users. The second and third models are more granular and are based on less frequent features, such as URL sharing patterns and hashtag sequences. These models cover only a thousand and three hundred users, respectively.

A comparison of the three models in terms of accuracy, precision and recall demonstrates that the URL-sharing model performs best. With an accuracy of 85%, precision of 83.33% and recall of 90.91%, it significantly outperforms both the friendship-relations model and the hashtag-sequences model. This might indicate that the bots used in Russian social networks tend to use URLs in their comments. Specifically, they seem to do so to promote Telegram channels, VKontakte groups and YouTube videos. Exploration of these sources of information reveals that these channels, groups and videos are usually highly biased and expose a strong position, whether a pro-Russian or pro-Ukrainian.

The developed bot detection model provides insights into the influence of political bots on Russian social networks that align with previous research on political bot detection. Firstly, social bots seem to be influential actors on VKontakte. Their removal from the user network leads to a decrease in the influentialness of the whole network. Secondly, they amplify the sentiments expressed in the comments. After their removal from the user network, the amplitude of negative sentiments expressed in user comments lessens. Overall sentiment, however, slightly increases. Bots influence the overall user network sentiment by leaving strongly negative and positive comments. Potentially, in this way, they can manipulate social network users' emotions. This aligns with the findings of previous studies for other social networks and supports the idea that bot behaviour on VKontakte is similar to Twitter bots' behaviour.

To evaluate the efficiency of the models, crowdsourced labelling of three different database samples was used. The labelling process itself revealed interesting insights into human judgment of bots. The majority of respondents considered pictures, and especially avatar photos, when trying to identify whether a given VKontakte user is a bot. However, with the current progress in image generation algorithms, pictures should not be considered a reliable sign of a real human profile. Attracting attention to the problem of social political bots is crucial to make social network users aware of the various features that can be indicative of bots, such as frequent URL sharing.

The bot detection model was used as a base for a web application that allows any user on the Internet to bot-check a given VKontakte user and can increase awareness in the Russian-speaking community about the political social bots in OSNs. The web application is publicly available and exposes the results of this work to a wide audience. Not only can it help users

distinguish bots and humans on VKontakte, but it also attracts attention to the problem of social political bots and potentially can help decrease their influence in social networks and make Russian-speaking social platforms more transparent, objective and safe.

6 Conclusion and Future Work

6.1 Future work

Opportunities for future work arise naturally from the limitations of the current study. To advance the detection of political bots in Russian-language social networks, such as VKontakte, it is necessary to address these limitations and solve them.

Firstly, a promising research area is the development of sentiment analysis techniques for less widespread languages than English and Russian. For instance, in this study, a model for sentiment analysis of Ukrainian texts would be beneficial.

Secondly, a challenging task for future research arises from the absence of ground truth to train and evaluate the models. The scientific community has long known that the current evaluation and training techniques are not perfect. However, only one reliable method of obtaining the ground truth has been invented. This method is not applicable to all studies and was not used in this research. Thus, solving the problem of absence of ground truth might benefit many researchers.

Thirdly, it is possible to try and build the users' graph based on a different behavioural trace than the friendship connection, URL sharing or hashtag sequences. This can produce results varying from those that emerged during this study. For example, users' similarity is one of the traces that could be researched, although it would require significant computational resources to be applied to large user networks.

Lastly, improvements to the web bot detection tool present a potential direction for future scientific endeavours. Extending the database with more users, analysing new clusters, making more information available in the web interface and improving the system based on users' feedback might be beneficial for the end users and help to raise awareness of the presence of bots in the online social environment.

6.2 Conclusion

Identifying political bots in online social networks is crucial for society and businesses due to their omnipresence and significant influence on the online environment. By influencing the online environment, bots can manipulate OSN users' opinions and help malicious individuals spread their agenda according to their goals. This thesis concerns detecting and investigating the behaviour of VKontakte political bots related to the Russian-Ukrainian armed conflict of 2022.

This research offers several contributions to political bot detection on Russian social networks.

Firstly, it presents one of the first attempts to build a bot detection model suitable for the identification of political bots on VKontakte, a major Russian OSN. The model is graph- and group-based, unsupervised, and relies on clustering to uncover coordinated communities of users.

Secondly, this research provides insights into political bot behaviour in the VKontakte discourse around the Russian-Ukrainian armed conflict of 2022. Judging by the centrality characteristics of the user network, bots seem to be influential actors in the online community. Moreover, bots influence the sentiments expressed in the network by producing more emotional content than humans and primarily amplifying negative emotions.

Thirdly, as a result of this research, a web application was developed to allow users to check if a user on VKontakte is a bot or a real person. The web application might be a tool to increase the transparency of social networks and increase awareness of political bots in the online environment.

Future research on this topic can take several directions in order to extend this study and improve its results. Working with the limitations of this research, especially concerning the model evaluation and finding the ground truth, may present significant interest to the social bot researchers interested in exploring the specifics of the modern informational war on social networks. In this emerging area of study, hopefully, this paper will become a foundation for future endeavours to uncover political manipulations, educate social network users and help people form unbiased and objective opinions without the influence of malicious social bots.

Bibliography

[ABS16] A. S. Alymov, V. V. Baranjuk, and O. S. Smirnova. "Detection of bot programs that mimic the behavior of people in the social network" Vkontakte"". In: *International Journal of Open Information Technologies* 4 (2016), pp. 55–60.

[AO20] Izzat Alsmadi and Michael J. O'Brien. "How many bots in Russian troll tweets?" In: *Information Processing & Management* 57 (2020), p. 102303.

[BE07] Danah M. Boyd and Nicole B. Ellison. "Social network sites: Definition, history, and scholarship". In: *Journal of computer-mediated Communication* 13 (1 2007), pp. 210–230.

[Cre+17] Stefano Cresci et al. "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race". In: *Proceedings of the 26th international conference on world wide web companion*. Proceedings of the 26th international conference on world wide web companion. 2017, pp. 963–972.

[Cre+20] S. Cresci et al. "Emergent properties, models, and laws of behavioral similarities within groups of Twitter users". In: *Computer Communications* 150 (2020), pp. 47–61.

[Cre20] Stefano Cresci. "A decade of social bot detection". In: *Communications of the ACM 63 (10)* 63 (2020), pp. 72–83.

[CSD22] Maurantonio Caprolu, Alireza Sadighian, and Roberto Di Pietro. "Characterizing the 2022 Russo-Ukrainian Conflict Through the Lenses of Aspect-Based Sentiment Analysis: Dataset, Methodology, and Preliminary Findings". In: *arXiv preprint arXiv:2208.04903* (2022).

[Cur22] Linda Curika. *What is the amount of automated activity on VKontakte?* 2022. URL: `https://www.linkedin.com/pulse/new-stratcom-coe-report-analyses-robotic-networks-vk-linda-curika/` (visited on 09/20/2022).

[De +11] Pasquale De Meo et al. "Generalized louvain method for community detection in large networks". In: *2011 11th international conference on intelligent systems design and applications*. IEEE. 2011, pp. 88–93.

[FA20] M. Fazil and M. Abulaish. "A socialbots analysis-driven graph-based approach for identifying coordinated campaigns in Twitter". In: *Intelligent & Fuzzy Systems* 38(3) (2020), pp. 2961–2977.

[GK22] Florian Gallwitz and Michael Kreil. "Investigating the Validity of Botometer-based Social Bot Studies". In: *arXiv preprint arXiv:2207.11474* (2022).

[Gri+17] Christian Grimme et al. "Social bots: Human-like by means of human control?" In: *Big data* 5.4 (2017), pp. 279–293.

[Hag+22] Loni Hagen et al. "Rise of the machines? Examining the influence of social bots on a political discussion network". In: *Social Science Computer Review* 40.2 (2022), pp. 264–287.

[Han+19] Simo Hanouna et al. "Sharp power in social media: Patterns from datasets across electoral campaigns". In: *Australian and New Zealand Journal of European Studies* 11 (2019).

[Hel+18] Todd C. Helmus et al. *Russian Social Media Influence: Understanding Russian Propaganda in Eastern Europe*. RAND Corporation, 2018.

[HKK16] Pascal Held, Benjamin Krause, and Rudolf Kruse. "Dynamic Clustering in Social Networks Using Louvain and Infomap Method". In: *2016 Third European Network Intelligence Conference (ENIC)*. 2016, pp. 61–68. DOI: 10.1109/ENIC.2016.017.

[Im+20] Jane Im et al. "Still out there: Modeling and identifying Russian troll accounts on Twitter". In: *12th ACM Conference on Web Science*. 12th ACM Conference on Web Science. 2020, pp. 1–10.

[KCK21] Maxim Kolomeets, Andrey Chechulin, and Igor V. Kotenko. "Bot detection by friends graph in social networks." In: *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* 12 (2021), pp. 141–159.

[Kol+21] Maxim Kolomeets et al. "Camouflaged bot detection using the friend list". In: *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). 2021, pp. 253–259.

[Kol+22] Maxim Kolomeets et al. "Experimental evaluation: can humans recognize social media bots?" In: *Preprint Communications ACM* (2022).

[Man22] Jeffrey Mankoff. *Russia's War in Ukraine: Identity, History, and Conflict*. Apr. 2022. URL: https://www.csis.org/analysis/russias-war-ukraine-identity-history-and-conflict.

[Niz+21] L. Nizzoli et al. "Coordinated Behavior on Social Media in 2019 UK General Election." In: *ICWSM* (2021). Query date: 2022-08-31 00:54:03. URL: https://ojs.aaai.org/index.php/ICWSM/article/download/18074/17877/21569.

[Pac+20] Diogo Pacheco et al. "Uncovering coordinated networks on social media". In: *arXiv preprint arXiv:2001.05658* 16 (2020).

[Par+22] Chan Young Park et al. "VoynaSlov: A Data Set of Russian Social Media Activity during the 2022 Ukraine-Russia War". In: *arXiv preprint arXiv:2205.12382* (2022).

[Pas+20] Javier Pastor-Galindo et al. "Spotting political social bots in Twitter: A use case of the 2019 Spanish general election". In: *IEEE Transactions on Network and Service Management* 17 (2020), pp. 2156–2170.

[Per+22] Paulo Pereira et al. "Russian-Ukrainian war impacts the total environment". In: *Science of The Total Environment* 155865 (2022).

[Rhe21] Andreea Musulan Ludovic Rheault. "Efficient detection of online communities and social bot activity during electoral campaigns". In: *Journal of Information Technology & Politics* (2021), pp. 324–337.

[Rio+18] Daniel Riofrio et al. "Tracking Elections: our experience during the presidential elections in Ecuador". In: *arXiv preprint arXiv:1807.06147* (2018).

[Rit] Luís Rita. *Louvain Algorithm*. URL: https://towardsdatascience.com/louvain-algorithm-93fde589f58c (visited on 09/10/2022).

[Ros+19] Sippo Rossi et al. "Detecting and analyzing bots on Finnish political twitter". Aalto University, 2019.

[Ros+20] Sippo Rossi et al. "Detecting political bots on Twitter during the 2019 Finnish parliamentary election". In: *Proceedings of the 53rd Hawaii international conference on system sciences*. Proceedings of the 53rd Hawaii international conference on system sciences. 2020.

[Sán+16] Daniel López Sánchez et al. "Twitter user clustering based on their preferences and the Louvain algorithm". In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer. 2016, pp. 349–356.

[Sim22] SimilarWeb. *SimilarWeb vk.com*. 2022. URL: https://www.similarweb.com/ru/website/vk.com/%5C#geography (visited on 09/18/2022).

[Sta21] Statista. *Leading social media platforms in Russia in 3rd quarter 2021, by penetration rate*. 2021. URL: https://www.statista.com/statistics/867549/top-active-social-media-platforms-in-russia/.

[Sta22a] Research Statista. *Number of civilian casualties in Ukraine during Russia's invasion verified by OHCHR as of March 5, 2023*. Aug. 2022. URL: https://www.statista.com/statistics/1293492/ukraine-war-casualties/.

[Sta22b] Research Statista. *Number of civilian deaths related to the Russia-Ukraine conflict from 2014 to 2021*. Apr. 2022. URL: https://www.statista.com/statistics/1293409/civilian-deaths-related-to-russia-ukraine-conflict/.

[Stu+19] Denis Stukal et al. "For Whom the Bot Tolls: A Neural Networks Approach to Measuring Political Orientation of Twitter Bots in Russia". In: *SAGE Open* (2019).

[Tim] The Moscow Times. *Gazprom Gains Control of Russia's Top Social Network*. URL: https://www.themoscowtimes.com/2021/12/03/gazprom-gains-control-of-russias-top-social-network-a75724 (visited on 09/12/2022).

[VLR19] Valeria Vasilkova, Natalia Legostaeva, and Vladimir Radushevsky. "Topical landscape of bot space on the social media platform Vkontakte". In: *Sociology of communication* 22 (2019), pp. 202–245.

[Wis] Jason Wise. *Twitter bot accounts: how many bots are on Twitter in 2022?* URL: https://earthweb.com/how-many-bots-are-on-twitter (visited on 07/22/2022).

[Woo16] Samuel C Woolley. "Automating power: Social bot interference in global politics". In: *First Monday* (2016).

[Zah22] Max Zahn. *A timeline of Elon Musk's tumultuous Twitter acquisition attempt*. July 2022. URL: https://abcnews.go.com/Business/timeline-elon-musks-tumultuous-twitter-acquisition-attempt/story?id=86611191.

A List of user features retrieved from VKontakte

Table A.1: List of user features retrieved from VKontakte

Feature	Description	Data source
created_at	Date and time when user created the profile.	FOAF response
timezone	Timezone of the profile.	
followee_rate	Number of followees of this user.	
follower_rate	Number of followers of this user.	
follower_to_followee	Ratio between followers number and followee number.	Calculation: follower_rate / followee_rate
vk_age	Number of days from account creation until 1st August.	Calculation: 01.08.2022 - created_at
first_name	User's first name.	VK API response
last_name	User's last name.	
photo_50	URL of a 50x50px profile picture.	
photo_100	URL of a 100x100px profile picture.	
screen_name	Short nickname of the user.	
is_closed	If the page is hidden by privacy settings.	
sex	User's sex.	
vk_id	User's ID.	
verified	If VKontakte moderators verified the user account with documents proving it is a real person.	
deactivated	If a user account is either banned or deleted.	

Declaration of Academic Integrity / Eidesstattliche Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe und dass alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind. Mit der aktuell geltenden Fassung der Satzung der Universität Passau zur Sicherung guter wissenschaftlicher Praxis und für den Umgang mit wissenschaftlichem Fehlverhalten vom 31. Juli 2008 (vABlUP Seite 283) bin ich vertraut. Ich erkläre mich einverstanden mit einer Überprüfung der Arbeit unter Zuhilfenahme von Dienstleistungen Dritter (z.B. Anti-Plagiatssoftware) zur Gewährleistung der einwandfreien Kennzeichnung übernommener Ausführungen ohne Verletzung geistigen Eigentums an einem von anderen geschaffenen urheberrechtlich geschützten Werk oder von anderen stammenden wesentlichen wissenschaftlichen Erkenntnissen, Hypothesen, Lehren oder Forschungsansätzen.

Passau, 9. März 2023

Valeriia Stromtcova

I hereby confirm that I have composed this scientific work independently without anybody else's assistance and utilising no sources or resources other than those specified. I certify that any content adopted literally or in substance has been properly identified. I have familiarised myself with the University of Passau's most recent Guidelines for Good Scientific Practice and Scientific Misconduct Ramifications from 31 July 2008 (vABlUP Seite 283). I declare my consent to the use of third-party services (e.g., anti-plagiarism software) for the examination of my work to verify the absence of impermissible representation of adopted content without adequate designation violating the intellectual property rights of others by claiming ownership of somebody else's work, scientific findings, hypotheses, teachings or research approaches.

Passau, 9. März 2023

Valeriia Stromtcova