Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1161–1166
DOI: 10.15439/2023F2738
ISSN 2300-5963 ACSIS, Vol. 35

Using graph solutions to identify "troll farms" and fake news propagation channels

Patryk Sulej
Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
Email: patryksulej2@gmail.com

Krzysztof Hryniów
0000-0001-8044-3925
Institute of Control and Industrial Electronics,
Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
Email: krzysztof.hryniow@pw.edu.pl

***Abstract*—This paper addresses the issue of fake news detection, with a particular focus on solutions derived from graph theory. It covers identifying channels, which are sources of fake news, and identifying users spreading false information, considering users deliberately misleading their audience, forming clusters called 'troll farms'. It proposes a solution using graph theory, which includes classifying users based on the social context extracted in graph centrality measures built from user interactions or networks built from followers on the social network Twitter. The solution includes not only the identification of trolls but also potential unintentional users spreading false information, users exposed to false information, or automated scripts spreading information (bots). Thorough research on the efficiency of different features and classifiers is conducted on MIB and FakeNewsNet datasets. Conducted research confirms general conclusions from previous studies and offers some improvements.**

I. Introduction

WITH the rapid increase in accessibility to information caused by the development of the Internet, it has become much easier to manipulate and spread false information to any audience. Social media platforms have changed the way journalism can be conducted in the 21st century, causing anyone to be able to report on events to the masses. What is most commonly considered fake news is, in a broad sense, information that is not true, or in a more specific purpose, information that has been made available to mislead the recipient [1].

Information portals or social media enable targeting any audience, which, if used appropriately, can influence public sentiment and impact countries' internal politics. It severely threatens a nation's and its citizens' stability and internal security. Examples of such events include the 2016 US presidential election campaigns, during which 20 of the most popular manipulated posts generated more shares and comments than 19 of the most prominent news sites [2]. 'Trolling' can be defined as deviant, malicious, anti-social behavior aimed at destroying a conversation or creating conflict. The key features of this activity are deception, aggression, and negative disruptive actions, and the measure of success is to gain as much audience attention as possible [3].

An example of how vital the information domain is can be seen in the actions taken by Russia and Ukraine during the Russian-Ukrainian war that began on 24 February 2022. Building public support for an invasion of a neighboring country using manipulative techniques and a wide range of information channels preceded the Russian Federation's attack on Ukraine [4]. This action also targeted the rest of the world – using messengers such as Telegram to release posts or videos distorting the picture of reality to present the Russian view of the conflict and gain support for its actions. While most Western countries did not succumb to disinformation, the manipulation work carried out domestically in the Russian Federation served its purpose and convinced most Russians that the war was necessary and consolidated citizens around the authorities.

II. Methods of fake news detection

Methods for detecting false information are classified as content-oriented, social context-oriented, and graph-based [5].

A. Content-oriented methods

Methods that use fact-checking, i.e., comparing the thesis presented in the news with external sources, are called knowledge-oriented methods. Manual fact-checking is poorly scalable and manpower-intensive. However, it allows for creating valuable datasets for developing automated solutions such as FakeNewsNet [6]. Fact-checking using 'crowd-sourcing' has a high risk of obtaining biased results, but it is better scalable than the expert method [1].

Style-oriented methods are similar to knowledge-oriented methods, but in this case, the aim is not to assess the content's veracity but to extract the author's intentions and determine whether it was to mislead the audience [1].

Content-oriented methods also include linguistic analysis of the text [7]. It is based on analyzing the syntax and semantics of a sentence by extracting features that distinguish false information from accurate information, such as length of statements, word embedding, lexical context, discourse level, etc. [5]. This solution works in the case of longer forms of expression, but in social media, extracting these features proves difficult, or there are too few to determine the veracity of such information.

B. Social context-oriented methods

One method used is to analyze the life of information on the web. It allows one to observe how it evolves with

each sharing and how information changes to form a 'rumor.' Analyzing the life cycle of such information over a period of time allows us to understand the diffusion patterns of rumors over time. Another way is to assess the information's veracity by analyzing the source's credibility [1][5]. The third popular solution for identifying fake news is to analyze the networks they form with other information, like social networks, friends, post sharing, interactions with other profiles, and profile data [1]. It allows for identifying the relationships between people spreading such information and extracting the characteristics of such interactions or profiles. An important aspect is the propagation pattern of such information, which differs between false and authentic information.

C. Graph-based solutions

Network and graph analysis is mainly based on studying features challenging to describe by standard data-averaging methods. In the case of graphs, there are often power relationships due to the uneven distribution of nodes or the high degree of links between data. Graph solutions allow the study of features such as the propagation speed of objects in the network, the relevance of individual nodes, or the way objects interact within the network and whether this can change.

Graph-based methods are used extensively in deep learning to detect internet trolls, fake news propagation channels, or fake news in general. Graph neural networks (GNNs) are characterized by the fact that they can encode the graph structure as well as the node features at the same time, which in the case of social networks or news propagation networks, dramatically increases the efficiency of classification [8][9].

To verify fake news, automatic fact-checking methods are often used, which consist of extracting facts from the content of the news and then comparing this fact with a knowledge base, the form of which can be a knowledge graph [1].

III. Dataset preparation and preprocessing

This paper decided to use graph centrality measures, which have served as features for machine learning algorithms. These measures were chosen because this area has yet to be fully explored despite some work on the subject.

Identifying 'fake' users based on a follower network is a method of detecting fake news based on social context. The source of false information can be identified in this way. When a new user arrives and 'adds' other users to his/her social network, there is a chance to identify whether he or she is an account that will spread false information. It creates a significant advantage because we can already take action, then – observe the user and start analyzing their content with other solutions to detect false information. References [10] and [11] examined follower networks and followed accounts using graph centrality measures. In addition, it was possible to classify online trolls from the 2016 US presidential campaign by creating a network of users who retweeted their posts [12]. Using graph algorithms, identifying Russian troll accounts extracted from a list provided by the US House Intelligence Committee from the 2016 US election was also feasible [13].

As part of this work, it was decided to use the centrality measures used in previous works, such as: centrality of agency, centrality of proximity (unnamed type), centrality of node degree (degree, in-degree, out-degree), PageRank centrality, centrality of eigenvector. In addition to this, the measures examined were: centrality of proximity (Wasserman-Faust), the centrality of harmonic closeness (harmonic closeness), ArticleRank, HITS (Hyperlink-Induced Topic Search).

A. Tools

Of the available tools for operating on graphs, it was decided to use Neo4j in the study because of the numerous previous uses of this tool for analyzing fake news and troll accounts [2][11][13]. This database is also fully adapted to operate on graphs. The research was performed on a computer with an Intel Core i7 7700HQ processor with 16GB RAM DDR5 and Google Colab. All collections were placed in the Neo4j database version 5.1.0. Additional libraries were used: APOC version 5.1.0 and Graph Data Science Library 2.2.5.

B. Datasets

The datasets used were those collected for the study of fake Twitter accounts [14]. This MIB dataset consists of five subsets: two sets of accounts run by humans (TFP and E13) and three sets of accounts with fake followers (INT, FSF, TWT). The data was collected before 2015. For machine learning, the collection was filtered, removing profiles with less than two edges due to their large number – they were considered noise. However, the complete set was used for feature extraction to capture the centrality features of all nodes as accurately as possible. In addition to the MIB collection, users extracted from the FakeNewsNet [6] collection were also used. This collection was created in 2018 and consisted of tweets spreading fake and real news, their retweets, the profiles of the users who sent them, and tweets from the users' timelines. The collection is based on manual fact-checking performed by the portals Gossipcop and Politifact.

Due to the known problems with the collection download and the Twitter limits [6][12][14][15], it was eventually possible to obtain 6 240 964 unique identifiers of users. Based on whether a profile was among the followers, or followers of an account that spread real or fake news, a label of true or false was assigned to that profile. Thus, profiles potentially at risk of seeing fake news were labeled as if they were spreading fake news. After filtering out the noise in the form of profiles that contained one or fewer relations and were irrelevant to the graph, 2 713 356 profiles were obtained. To speed up the Neo4j database feature extraction algorithms, once the collection was imported, the Random Walk with Restart algorithm was used to sample the collection at a ratio of 0.3. This algorithm preserves the structural features of the graph, which, in the case of centrality testing, is crucial for obtaining results close to the truth. Unfortunately, this procedure nevertheless introduced additional uncertainty into the study. The final result was 541 255 nodes labeled as potentially false and 272 743 as potentially genuine.

TABLE I: Size of follower datasets.

Dataset name	Number of profiles	Final number of profiles	Genuine/Fake accounts
TFP	240 961	198 621	Genuine
E13	996 438	147 955	Genuine
TWT	77 685	24 436	Fake
INT	57 266	17 578	Fake
FSF	20 173	5 914	Fake
FakeNewsNet #1	6 240 964	541 255	Fake
FakeNewsNet #2	-	272 743	Genuine
Sum	7 633 487	1 208 502	-

TABLE II: Size of user interaction sets from the FakeNewsNet skeleton and US Elections Trolls.

Dataset name	graphs (with fake news propagation)	profiles retweeting genuine news	profiles retweeting fake news
Politifact	314 (157)	18 042	23 012
Gossipcop	5 464 (2 732)	208 079	106 183
USElectionsTrolls	269 (269)	0	413
Sum	6 047 (3 158)	226 121	129 195

As the access to information about real troll accounts via the Twitter API was prevented, and the data contained in the Neo4j Sandbox about these accounts were small, another collection [1] from the GNN Fake News survey was used [8]. That survey used the FakeNewsNet dataset and provided the collection as a finished graph – the relationship between individual users who retweeted another user's post. The collection in this form contains much less memory because the original tweet identifiers have been mapped to unique numerical values starting from 0, and it does not contain additional information related to the user profile – it is a kind of skeleton. This processed collection yielded user profiles, with interaction in the form of retweeting a post. The original collection contained 425 842 profiles, but due to accounts being blocked, deleted, or unavailable, the authors obtained only 355 316. Tweet collection is a separate part of the MIB collection. It was created for the paper [15] on the study of spambots.

C. Selection of characteristics of user interaction and followers sets

To select features for machine learning algorithms, the following dependency measures were used: Pearson correlation, F-test, analysis of variance (F classifier), Mutual information, chi2 (chi-square test), tree classifier [16][17].

The feature selection analysis was started by determining the Pearson correlation matrix, identifying linearly dependent features, and then sifting them out. The selection was carried out on the complete set, with the awareness that some algorithms will show a linear relationship because they are similar in implementation – for example, closeness and harmonic closeness, or PageRank and ArticleRank.

The significance level of $\alpha = 0.1$ was assumed to reject the null hypothesis. Thus, for p > 0.1, we cannot reject the null hypothesis that the variables are independent. The choice to leave one of the two features was made when the linear Pearson correlation coefficient between the features exceeded the value of 0.3.

[1]https://github.com/safe-graph/GNN-FakeNews

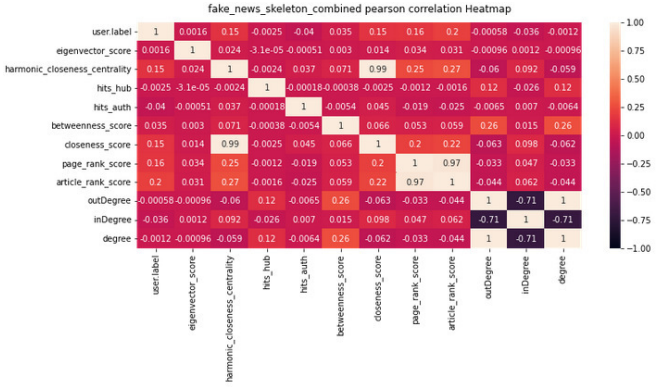

Fig. 1: Pearson correlation matrix of features extracted using graph algorithms for the FakeNewsNet skeleton - combined Politifact and Gossipcop sets.

Pearson correlation matrices were prepared for different cases: separate Politifact user interaction set, separate Gossipcop user interaction set, combined Politifact and Gossipcop user interaction set, MIB set of followers, and combined Politifact and Gossipcop set of followers. Results for different cases were similar. One of Pearson correlation matrices is shown in Figure 1. Based on them, selected features for user interaction sets were: eigenvector score, closeness score, hits auth, page rank, and inDegree. For the MIB set of followers and combined Politifact and Gossipcop set of followers, the following features were selected: eigenvector score, harmonic closeness score, hits auth, page rank.

IV. Classification

To roughly identify the classifiers that will bring the best effect, the extracted features were trained using the following algorithms:

- K-Neighbors Classifier, where k is set to n=3 by default;
- classifier with decision tree algorithm (Decision Tree Classifier);
- classifier with Random Forest Classifier, where the number of heuristic estimators was initially set at 300;
- adaptive boost classifier (AdaBoost Classifier);
- Gradient Boosting Classifier;
- Gaussian classifier with naive Bayes algorithm (GaussianNB);
- Linear Discriminant Analysis classifier;
- Quadratic Discriminant Analysis classifier;
- Support Vector Machines Classifier (SVC), with regularization parameter C=0.025, radial basis function kernel, and 5-fold cross-validation;
- Support vector classifier with support vector quantity control proposed by Bernhard Schölkopf (NuSVC - Nu Support Vector Machines Classifier) [18].

Algorithms with unspecified configurations used the default settings of the Sci-Kit Learn library. An initial test was carried out using the Accuracy index and the Log Loss function to determine the confidence with which the algorithm made the classification [19]. The sets were divided using the function

TABLE III: The results of individual algorithms measuring data dependence for the combined set of Politifact and Gossipcop (skeleton of the FakeNewsNet set)

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.020	0.929	0.335	0.237	0.627	0.002	0.002
harmonic closeness centrality	0.020	7925.474	0.000	0.101	0.750	0.148	0.046
hits hub	0.000	2.148	0.143	1.530	0.216	-0.002	0.000
hits auth	0.011	583.671	0.000	25.271	0.000	-0.040	0.016
betweenness score	0.009	438.670	0.000	13913.119	0.000	0.035	0.018
closeness score	0.021	8073.976	0.000	0.086	0.770	0.149	0.047
page rank score	0.225	9778.541	0.000	74.900	0.000	0.164	0.386
article rank score	0.259	14958.180	0.000	25.026	0.000	0.201	0.445
outDegree	0.011	0.121	0.728	7.679	0.006	-0.001	0.018
inDegree	0.048	472.832	0.000	7.679	0.006	-0.036	0.003
degree	0.031	0.495	0.482	15.358	0.000	-0.001	0.019

TABLE IV: The results of individual algorithms measuring data dependence for the MIB set of followers

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.245	1950.344	0.000	9.942	0.002	-0.037	0.001
harmonic closeness centrality	0.271	196599.696	0.000	8538.933	0.000	-0.352	0.353
hits hub	0.190	26842.827	0.000	981.298	0.000	-0.138	0.286
hits auth	0.248	693.709	0.000	3.032	0.082	0.022	0.036
betweenness score	0.002	2.082	0.149	4.20e9	0.000	0.001	0.000
closeness score	0.271	190243.421	0.000	7738.225	0.000	-0.347	0.289
page rank score	0.264	3580.209	0.000	22986.895	0.000	-0.051	0.000
article rank score	0.260	1308.763	0.000	708.467	0.000	-0.031	0.000
outDegree	0.114	51.639	0.000	3397591.733	0.000	-0.006	0.018
inDegree	0.178	11221.404	0.000	3.4e7	0.000	-0.089	0.007
degree	0.053	204.437	0.000	6795183.465	0.000	-0.012	0.010

TABLE V: The results of individual algorithms measuring data dependence for the FakeNewsNet set of followers

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.244	3560.221	0.000	19.645	0.000	-0.066	0.163
harmonic closeness centrality	0.263	2828.637	0.000	3.670	0.055	-0.059	0.168
hits hub	0.103	3.883	0.049	0.077	0.782	-0.002	0.069
hits auth	0.261	1850.662	0.000	9.523	0.002	-0.048	0.159
closeness score	0.260	1961.225	0.000	2.354	0.125	-0.049	0.174
page rank score	0.266	4.536	0.033	306.354	0.000	-0.002	0.136
article rank score	0.271	61.463	0.000	164.771	0.000	-0.009	0.131

TABLE VI: Stratified 10-fold cross-validation results for selected classifiers for the combined set of Politifact and Gossipcop.

Classifier	Mean Validation Accuracy	Mean Validation Precision	Mean Validation Recall	Mean Validation F1 Score
K-Neighbors	75.082	0.917	0.346	0.502
Random forest	80.714	0.812	0.611	0.697
AdaBoost	72.101	0.652	0.499	0.565
Gradient boosting	73.710	0.688	0.506	0.583

sklearn.model_selection.train_test_split from the sci-kit learn library, which split the set on a scale of 0.7 into training and test sets [17].

Finally, the following classifiers were subjected to further analysis: random forest, gradient boost, k-nearest neighbors, adaptive boost. These classifiers were subjected to a stratified 10-fold cross-validation study following a review of popular methods for testing the efficiency of classifiers [20]. Stratification is a good solution for unbalanced sets, and the K-fold method itself has already been used in previous works on this topic [10][11]. It is also widely used, and effective [21]. The study results are presented in Table VI.

Table VI shows that all models obtained a relatively low recall value, indicating many classifications of "fake" users as "real". A better result was obtained in the case of precision, which gives us information about how many "real" accounts were rated as "fake". Fewer false profiles in the set (Table II) could have contributed to obtaining a high value of the accuracy coefficient.

When detecting fake user accounts, it is essential to consider how much it will cost to recognize a user spreading accurate information when they are a "troll". This cost can be very high, making the built algorithm useless. Sometimes, however, the "forbearance" of the algorithm can be desirable.

The final proposed solution is a classifier based on the random forest algorithm, where the number of estimators n has been heuristically set to n=300. This classifier was tested on the set of Russian troll accounts described in Table II. This set consisted of 413 fake accounts and was used only as another measure of verifying the task's success. The final version of the random forest classifier learned from the Politifact and Gossipcop collections achieved an accuracy of 84.50%. This observation is consistent with previous conclusions for the set of low validity, but the obtained result is better than the tests would indicate.

A. Choosing a solution to detect fake propagation channels and bots by analyzing the network of followed users

The classifiers were studied for these sets by testing the best-performing algorithms using selected features. Studies for the

TABLE VII: Test results of different classifiers on a set of MIB followers using 10-fold cross-validation with stratification.

Classifier	Mean Val. Accuracy	Mean Val. Precision	Mean Val. Recall	Mean Val. F1 Score
KNN	99.822	1.000	0.998	0.999
Decision Tree	99.388	0.996	0.997	0.997
Random Forest	99.397	0.996	0.997	0.997
AdaBoost	98.626	0.996	0.988	0.992
Gradient Boosting.	99.388	0.996	0.997	0.997
Gaussian NB	90.731	0.927	0.973	0.949
Linear Disc. Anal.	92.734	0.925	1.000	0.961
Quadratic Disc. Anal.	90.615	0.926	0.972	0.948

TABLE VIII: Test results of various classifiers on a set of FakeNewsNet followers without using cross-validation. (70% training data, 30% test data)

Classifier	Accuracy	Precision	Recall	F1 Score	Log Loss
Decision Tree	77.454	0.832	0.828	0.830	7.784
Random Forest	82.035	0.816	0.941	0.874	0.346
Gradient Boosting	70.184	0.693	0.990	0.815	0.584
GaussianNB	66.330	0.669	0.975	0.794	0.815
KNN	77.484	0.814	0.857	0.835	2.710
AdaBoost	66.888	0.670	0.987	0.798	0.690
Linear Disc. Anal.	66.660	0.667	0.997	0.799	0.635
Quadratic Disc. Anal.	66.411	0.669	0.978	0.795	0.790

sets were performed using 10-fold cross-validation to better compare the results with those in other studies. In addition, the results of training performed on one set and then testing the model on a second set were also examined.

Table VII shows that the classifiers obtained high confidence and accuracy on the MIB set. This may be because it consisted of accounts generated by bots, which may have resulted in more significant differences between the characteristics. An important fact is that this set is already about ten years old, so the algorithms creating the bots could have been less advanced then. Similar high accuracy was achieved for all algorithms except Gaussian naive Bayes and linear and quadratic discriminant analysis.

Other studies based on measures of centrality have yielded for this set:

- in 2016 - precision 89.0%, accuracy 100% and validity 95% [10];
- in 2021 - precision, accuracy, and validity of 99.5%[11].

The extracted features for this set allowed us to obtain results similar to the work [11] where closeness centrality was introduced. At the same time, it can be seen that betweenness centrality, in this case, does not play a significant role in classifying "fake" users. It was also possible to obtain better results with the KNN classifier than previous works did with the random forest.

Worse algorithm efficiency results were obtained for the set of FakeNewsNet followers, presented in Table VIII. In this case, the random forest classifier was the best, achieving the highest accuracy and the lowest Log Loss. The decision tree algorithm, KNN, and gradient boost also achieved high scores. Worse results could be obtained because accounts were classified as fake or genuine only because they had a person who tweeted false information to their followers or followed. However, achieving such accuracy means that we can identify people who may be potentially unwitting spreaders of fake news, and according to research, they constitute a large part of fake news propagation channels [1].

TABLE IX: Test results of various classifiers learned on a set of FakeNewsNet followers and tested on a set of MIB followers without cross-validation. (70% training data, 30% test data)

Classifier	Accuracy	Precision	Recall	F1 Score	Log Loss
Decision Tree	34.155	0.772	0.368	0.498	22.753
Random Forest	87.018	0.894	0.969	0.930	0.668
Gradient Boosting	89.342	0.894	0.998	0.943	0.376
GaussianNB	8.762	0.000	0.000	0.000	5.609
KNeighbors	89.956	0.901	0.996	0.946	3.487
AdaBoost	90.237	0.903	0.998	0.948	0.683

TABLE X: Stratified 10-fold cross-validation results for selected classifiers for the combined set of FakeNewsNet and MIB followers.

Classifier	Mean Val. Accuracy	Mean Val. Precision	Mean Val. Recall	Mean Val. F1 Score
Random Forest	93.109	0.937	0.981	0.958
Gradient Boosting	87.988	0.875	0.994	0.930
KNN	91.647	0.942	0.955	0.949
AdaBoost	84.504	0.870	0.950	0.908

However, surprising results were obtained for the model trained on the FakeNewsNet set and tested on the MIB set containing bots. The results presented in Table IX show that both the decision tree algorithm and the naive Bayes classifier performed much worse in this case than before. An interesting result was obtained in the case of the adaptive gain algorithm, which turned out to be the best in terms of precision and in terms of accuracy. The gradient boost and random forest algorithms also performed well. The KNN method obtained a relatively high value of the Loss Log coefficient. This result was probably obtained because the MIB set profiles were relatively easy to detect. The model built on FakeNewsNet seems to be quite effective in this case. In the reverse situation, when the MIB model was used on the FakeNewsNet set, worse results were obtained – it can be assumed that the model built on this set will have a lower generalization ability.

The random forest, gradient boost, KNN, and adaptive boost classifiers were tested on a combined set of FakeNewsNet and MIB followers to maximize the efficiency. Table X shows the result of testing the effectiveness of classifiers using 10-fold cross-validation with stratification.

The obtained values are slightly worse than those obtained in the research from 2021 [11] on the exclusive MIB set. However, the MIB set allowed us to build a classifier and significantly lower ability to generalize in detecting fake users, in contrast to the set of FakeNewsFollowers obtained in this work. Building a classifier based on both sets significantly increases the generalization capabilities of the classifier.

Ultimately, the best overall results were obtained for the random forest, which confirms previous studies. At the same

time, other algorithms have also been shown to be highly effective. Very high accuracy was obtained for the gradient boost, which may be beneficial in the case when a maximum "raw" classifier is needed in detecting fake users, even at the cost of considering some genuine users as fake.

V. Conclusions

The article presents graph techniques for detecting false information. An essential aspect of detecting fake news is combining knowledge from many disciplines and data from different contexts to get better results. Combining several methods gives better results, but creating a complete system that classifies information as genuine and false using content and social context analysis is time-consuming and complicated. Studying individual techniques of a complex solution, such as the one presented in the paper, requires much time and collecting appropriate training data for machine learning algorithms.

The problem of classification presented in both cases, for the analysis of connections between users based on retweeting posts and based on followers, turned out to be a complicated issue. In the case of the user interaction network, it was impossible to build an effective classifier to solve the problem. However, we created a classifier that dealt with accounts of Russian trolls from the 2016 US elections quite effectively, proving that research in this direction should be continued.

It is more difficult to determine whether a user is part of a fake news channel based on what users they retweet. In further research, the set should be enlarged with additional samples, more work should be done to remove potential outliers, and the set should be better balanced to avoid overfitting. An important area for improvement is set normalization, model regularization, and parameter tuning.

Classifier tests in the case of the follower network essentially confirmed the conclusions regarding the effective operation of the random forest from previous studies [10][11]. It turned out, however, that the KNN classifier on the same set of MIB followers achieved better results than the random forest used in previous studies. It is an important finding, considering that learning this algorithm took less time than in the case of a random forest for n=300 estimators. Also, learning on the set of FakeNewsNet followers and validation on the MIB set was reasonably practical – although it could have been more reliable among the algorithms, obtaining a considerable value of the Log Loss coefficient.

References

[1] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, 9 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3395046

[2] C. Silverman, "This analysis shows how viral fake election news stories outperformed real news on facebook. buz- zfeed news." [Online]. Available: https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[3] H. F. Gylfason, A. H. Sveinsdottir, V. Vésteinsdóttir, and R. Sigurvinsdottir, "Haters gonna hate, trolls gonna troll: The personality profile of a facebook troll," *Int J Environ Res Public Health*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8199376/#B1-ijerph-18-05722

[4] M. Marek, "Russian information war: the activities of the russian propaganda apparatus in the context of the war in ukraine (as of the first half of march 2022)," *Bezpieczeństwo teoria i praktyka*, 2022. [Online]. Available: https://btip.ka.edu.pl/btip-2022-nr3/

[5] S. Hangloo and B. Arora, "Fake news detection tools and methods – a review," *International Journal of Advance and Innovative Research*, 6 2021. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2112/2112.11185.pdf

[6] K. Shu, D. Mahudeswaran, D. L. Suhang Wang, and H. Liu, "Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media," 2018. [Online]. Available: https://arxiv.org/abs/1809.01286

[7] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news." The 2nd International Workshop on News and Public Opinion at ICWSM, 2017. [Online]. Available: https://arxiv.org/abs/1703.09398

[8] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, "User preference-aware fake news detection," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021. [Online]. Available: https://arxiv.org/abs/2104.12259

[9] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," vol. abs/1902.06673, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918318210

[10] A. Mehrotra, M. Sarreddy, and S. Singh, "Detection of fake twitter followers using graph centrality measures," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016. doi: 10.1109/IC3I.2016.7918016 pp. 499–504.

[11] Y. Zhao and J. Weber, "Detecting fake users on social media with a graph database," vol. 12, 10 2021. [Online]. Available: https://doi.org/10.18357/tar121202120027

[12] A. Badawy, E. Ferrara, and K. Lerman, "Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 8 2018. [Online]. Available: https://doi.org/10.1109\%2Fasonam.2018.8508646

[13] W. Lyon, "The story behind russian twitter trolls: How they got away with looking human – and how to catch them in the future," 3 2018. [Online]. Available: https://neo4j.com/blog/story-behind-russian-twitter-trolls

[14] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923615001803

[15] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race." Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 963–972. [Online]. Available: https://doi.org/10.1145/3041021.3055135

[16] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLOS ONE*, vol. 9, no. 2, pp. 1–5, 02 2014. doi: 10.1371/journal.pone.0087357. [Online]. Available: https://doi.org/10.1371/journal.pone.0087357

[17] scikit learn.org, "Api reference." [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection

[18] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, p. 1207–1245, 5 2000. [Online]. Available: https://doi.org/10.1162/089976600300015565

[19] I. J. Good, "Rational decisions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952. [Online]. Available: http://www.jstor.org/stable/2984087

[20] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance." *Journal of machine learning research*, vol. 11, no. 6, 2010. [Online]. Available: https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf

[21] P. Refaeilzadeh, L. Tang, and H. Liu, "On comparison of feature selection algorithms," in *Proceedings of AAAI workshop on evaluation methods for machine learning II*, vol. 3, no. 4. AAAI Press Vancouver, 2007, p. 5. [Online]. Available: https://www.aaai.org/Library/Workshops/2007/ws07-05-007.php