Detecting of Anti-Ukrainian Trolling Tweets

Kostiantyn Vyrodov, Anastasiya Chupryna and Ruslan Kotelnykov

Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine

Abstract

The research aims to analyze the effectiveness of the modern machine-learning models usually used for data classification to detect anti-Ukrainian trolling tweets on Twitter. This research was conducted based on 6000 manually gathered tweets. The gathered dataset is divided into training and validation subsets of 75% and 25%, respectively. Also, it consists of 3000 pro-Ukrainian tweets and 3000 anti-Ukrainian tweets. Specific conditions of experiments, models, performance metrics, platform, type of learning, and classification efficiency indicators are determined. SVM, Decision Tree, Multinomial Naive Bayes, and Logistic Regression models are trained using supervised machine learning on the colab research google platform. The evaluation is done by analyzing famous classification metrics, such as accuracy, precision, recall, and F1 score. Finally, the results of experiments are given, along with conclusions and practical recommendations on using machine learning models.

Keywords

Machine Learning, SVM, Decision Tree, Multinomial Naive Bayes, Logistic Regression, Twitter, Bot, Troll, NLP

1. Introduction

On 24 February 2022, Russia invaded Ukraine in a major escalation of the Russo-Ukrainian War, which began in 2014. However, Russian aggression is not limited to the battleground but includes cyberattacks and PSYOPS (Psychological Operations) in social media.

PSYOPS are operations to convey selected information and indicators to audiences to influence their emotions, motives, objective reasoning, and ultimately the behavior of governments, organizations, groups, and individuals. Today, social media platforms are perfect for performing PSYOPS via troll accounts spreading misleading information. A troll is a person who posts or makes inflammatory, insincere, digressive, extraneous, or off-topic messages online with the intent of provoking others into displaying emotional responses or manipulating others' perceptions.

Twitter is a popular social network that the Russian government widely uses to spread disinformation about the war in Ukraine, spoil the Ukrainian reputation, and convince Ukrainian allies to stop their support. Therefore, detecting and eliminating troll accounts and their fake trolling content will positively affect the security of Ukrainians and complicate the execution of PSYOPS for the aggressor.

Machine learning is one of the approaches that can be used to identify trolling content on Twitter. This research aims to gather an up-to-date dataset related to the Russia-Ukrainian war on the Twitter platform and to set up experiments in order to determine trolling content using the widely used machine learning models, evaluate the effectiveness of each model within specific conditions, and formulate recommendations on the practical application of machine learning techniques and methods to solve this type of problem.

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine

EMAIL: kostya.vyrodov@gmail.com (K. Vyrodov); anastasiya.chupryna@nure.ua (A. Chupryna); ruslan.kotelnykov@nure.ua (R. Kotelnykov)

ORCID: 0009-0006-1746-2334 (K. Vyrodov); 0000-0003-0394-9900 (A. Chupryna); 0000-0003-2413-1809 (R. Kotelnykov)

2. Related works

Detecting trolling bots is not easy because anyone can post trolling content online. Currently, more and more researchers are devoted to developing models and technologies for protecting people from cyberbullying (trolling) [1-3]. In this respect, paper [1] presents the results of the analysis of cyberbullying in social networks, paper [2] describes a transfer learning model for training neural networks to recognize the facts of cyberbullying in social networks, and paper [3] proposes an effective model for detecting emotions in messages and comments from social networks.

The paper [4] proposed an integrated model to classify cyber harassment in social networks. The paper [5] describes behavior-based machine-learning approaches for identifying government-sponsored Twitter trolls. The paper [6] proposes and presents a model for detecting trolls based on user sentiment analysis, including the results of experiments proving this statement. The paper [7] describes the detection of cyber trolls using a model for extracting word embeddings (including hashtags) from tweets to identify groups of interest. The works [8, 9] provide an up-to-date review of models and algorithms for detecting farms and networks of trolls, Twitter bots, and their posts when interfering with networks at the state level. The paper [10] shares models and algorithms for detecting facts of collusion between retweets. The paper presents the results of the analysis, detection, and characteristics of such trolls and messages. The work [11, 12] provides an interesting model for establishing parallels and transferring technologies from electronic warfare to detecting and combating fake news, trolls, and troll farms. The paper [13] considers the topical issue of the online trolling ecosystem. Since trolling is integral to the functioning of modern social networks, models are proposed for detecting trolling facts along with interesting assessments, analyses, and recommendations for practical application.

The analysis of the current state of this problem shows that the vast majority of research is devoted to analyzing information on Twitter based on the use of neural networks and ML. Papers [14, 15] propose emotion recognition results on Twitter using the Unison application model. In addition, the papers provide the results of comparable studies and learning outcomes. The paper [16] uses a multi-aspect neural network Attentional Graph to determine the user's location in a social network. The analysis of modern publications shows an excellent prospect for using neural networks, not only for the analysis of textual information but also for the effect of (graphic) accumulation [17–19], mainly people's faces and emotions. The paper [20] presents the results of a study on the imitation and recognition of sarcasm on Twitter. The work [21] presents the results (models and algorithms) for detecting and extracting social events from Twitter based on the BiLSTM-CRF model. The paper [22] presents the results of the effects of political polarization of opinions (posts) in social networks based on the use of neural networks. The paper [23] presents the results of detecting rumors in social networks using transformer-type models. The paper [24] proposed a new algorithm that was called the "multilevel tweet analyzer" (MLTA). This algorithm allows text to be graphically represented in social networks using multi-layer networks (MLN) in order to obtain better results of coding relationships between independent sets of tweets.

The development of modern representation models is no less important for the combination and presence of cyberbullying facts estimation in e-learning and some other systems [25, 26].

A study analyzing government-sponsored trolls related to the Russian troll farm found that usually trolling bots create a small portion of destructive content, such as posts or comments, and heavily spread them by retweeting and copy-pasting the same information within a specific period [27]. Existing Twitter bot detection methods can be grouped into feature-based, text-based, and graph-based methods [28].

The idea of feature-based methods is to discover features from user information and utilize machine learning classification algorithms to detect bots. Researchers extract properties from users' metadata, their follow relationships, and tweets, including various time patterns. The paper [29] presents results where researchers achieved 87% accuracy in detecting Twitter bots using different machine-learning methods on a dataset containing metadata about Twitter profiles. However, bot owners are increasingly aware of discovered features allowing others to identify bot accounts, so new bots try to imitate other behavior to evade detection. Subsequently, engineers implementing this approach for bot detection have to keep track of bot evolution to keep their models actual.

Graph-based methods treat Twitter as graphs using concepts from network science. This approach adopts neural graph networks, heterogeneous graph neural networks, and node representation learning to detect Twitter bots. For example, a group of researchers from Xi'an Jiaotong University proposed TwiBot-22, a graph-based Twitter bot detection benchmark that presents a comprehensive dataset, providing diversified entities and relations on the Twitter network. They re-implemented 35 Twitter bot detection baselines, evaluated them on nine datasets, and achieved about 80% accuracy [30].

Text-based methods utilize techniques in NLP to detect trolling bots based on tweets. Under the hood, the methods use word embeddings, recurrent neural networks, and pre-trained language models. Since trolling content is primarily textual and usually represented as a comment or a post containing hostile language, employing a linguistic and sentiment analysis is a good approach for detecting trolling content.

The paper [31] shares the results of applying domain-adaptation techniques for sentiment analysis of textual content in online forums. The researchers achieved around 70% in detecting trolls. In the paper [32], researchers evaluated the sentiments of posts and other metadata from trolling posts and were able to detect Twitter trolls more than 76% of the time. C.J. Hutto and Eric Gilbert presented VADER [33], a simple rule-based model for general sentiment analysis. Utilization of the VADER in combination with sentiment, aggression, lexical, and syntactic textual features to determine whether a tweet is meant to troll or not achieved 88% accuracy when tested with the Kaggle Twitter cyber-trolls dataset [34, 35]. Todor Mihaylov and Preslav Nakov developed two classifiers for detecting "sponsored trolls" trying to manipulate the public's opinion and another for detecting "individual trolls" trying to provoke negative emotions. They combined sentiment analysis with metadata of trolling posts (information about the publication time) and achieved 82% accuracy [36].

3. Methods and materials

Consider input data, used methods and conditions for experiments and metrics to understand which model demonstrates better results.

3.1. Data description

In this study, existing Twitter datasets with already identified trolling users and trolling tweets (e.g., the IRA troll dataset or the Dataset of Russian trolling tweets for detection of cyber-trolls [37]) were not used because they are not directory related to the context of the Russia-Ukrainian war.

The data set, which was used for the research, consists of raw new tweets and labels specifying whether a tweet is anti-Ukrainian or not. The tweets were gathered via Twitter API and filtered by one of the similar keywords: "zov", "nazis", "azov", "russia is a terrorist state" or "putin war crimes" including different hashtags such "#RussiaInvadedUkraine" or "#ZOV". Such keywords were selected to find tweets where a user wanted to deliberately create an association with one of the sides in this war. Each tweet was manually labeled as a pro-Ukrainian or an anti-Ukrainian one by the researcher. All tweets were gathered via a JS script and saved into a Comma Separated Value (CSVs) file that could be easily imported into an ML model.

The dataset contains 6000 items. There are 3000 of anti-Ukrainian tweets and 3000 of pro-Ukrainian tweets. Table 1 demonstrates two samples from the data set.

Table 1
Data set samples

TweetId	Text	Label
1593089288404873217	Zelensky is a war criminal and NATO is his enabler.	AntiUkrainian
1620546714724880386	Ministry of Defense 🇺🇦showed the work of the Polish 🇵🇱"Crabs" at the front. #Ukraine #Poland #StopRussia	ProUkrainian

The "TweetId" column represents an id of the tweet and can be used in future research to get extra data about the tweet. "TweetId" is represented as a number containing 19 digits.

The “Text” column is a raw tweet congaing a maximum of 280 characters, and the “Label” column points to the category of the concrete sample. The “Text” can contain arbitrary characters and words, including emojis, links, or hashtags.

The dataset was split into two parts to analyze content and build the frequency distribution charts based on used words. The first part contained anti-Ukrainian tweets and the second contained pro-Ukrainian tweets. Each tweet was split into words and filtered from stop words and non-alphabetical symbols. Figure 1 displays the distribution of the words in anti-Ukrainian tweets, and Figure 2 displays the distribution in pro-Ukrainian tweets.

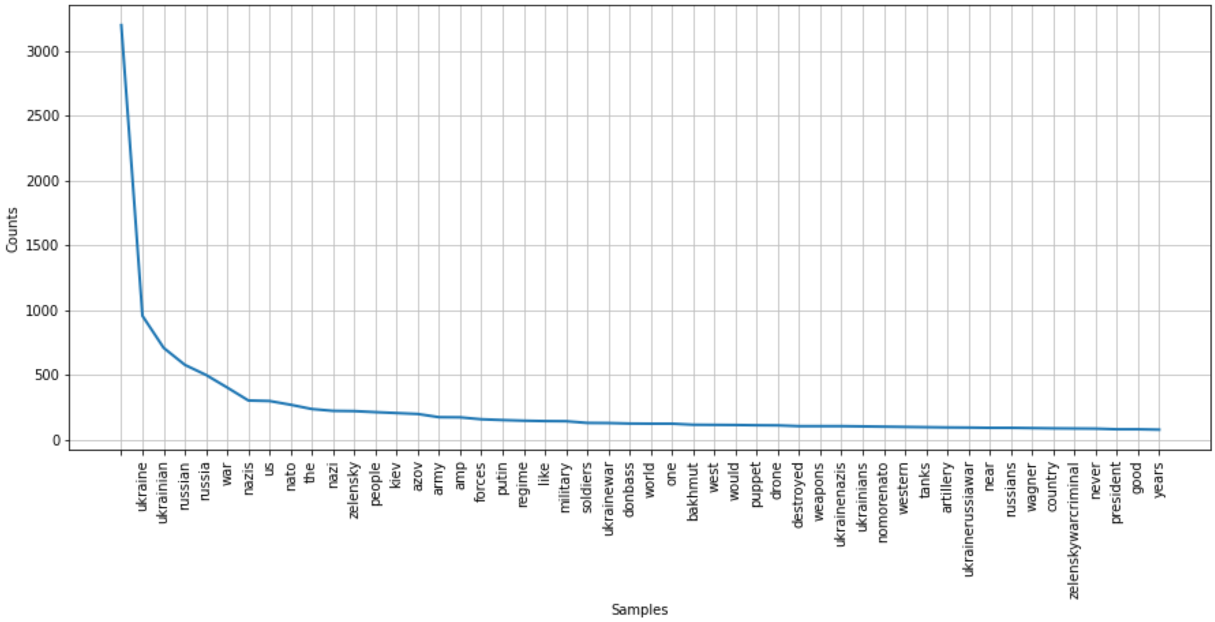


Figure 1: Words distribution in anti-Ukrainian tweets

It is possible to see in Figure 1 that the most popular words in anti-Ukrainian tweets are “Ukraine”, “Ukrainian”, “russian”, “Russia”, “war”, “nazi”, “NATO”, “zelensky”. In addition, these tweets contain specific for this category words, such as “zelenskywarcrime” or “ukrainenazis”.

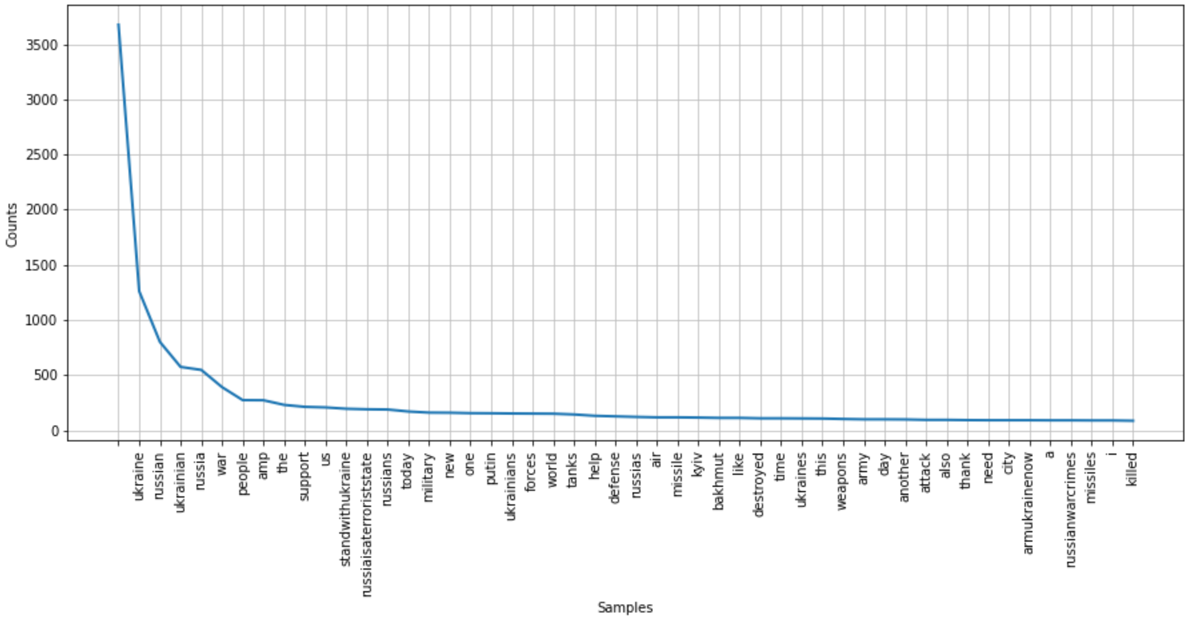


Figure 2: Words distribution in pro-Ukrainian tweets

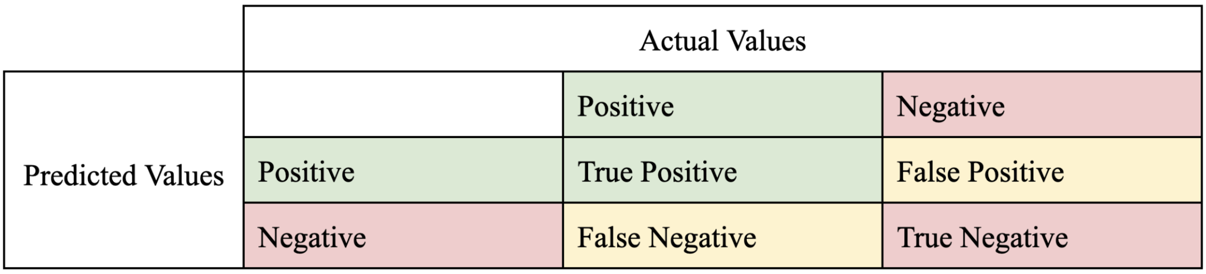
Figure 2 displays that the most popular words in pro-Ukrainian tweets are “Ukraine”, “russian”, “ukrainian”, “russia”, “war”, “people”, “support”. These tweets contain specific for this category words as well, such as “russiaisaterrorisstate” or “standwithukraine”.

3.2. Machine learning models validation and metrics/efficiency indicators

Machine learning models are built based on feedback from evaluated performance metrics that help to understand whether a model meets requirements.

There are different metrics in the AI Industry, such as recall or precision, helping to evaluate the performance of a model. This research will use accuracy, precision, recall, and F1 metrics that are derived from the confusion matrix.

A confusion matrix (Figure 3) is a tabular structure that helps visualize the performance of classifiers. Each column in the matrix represents classified instances based on predictions, and each row of the matrix represents classified instances based on the actual class labels.



		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 3: Confusion matrix

True Positive (TP) indicates the number of correct hits or predictions for our positive class. False Negative (FN) indicates the number of instances we missed for that class by predicting it falsely as the negative class.

False Positive (FP) is the number of instances we predicted wrongly as the positive class when it was not.

True Negative (TN) is the number of instances we correctly predicted as the negative class.

Accuracy is defined as the overall accuracy or proportion of correct predictions of the model, which can be depicted by the formula (1) where we have our correct predictions in the numerator divided by all the outcomes in the denominator.

$$Accuracy = \frac{TP + TN}{TP\ +\ FP\ +\ FN\ +\ TN}. \quad (1)$$

Precision is defined as the number of predictions made that are correct or relevant out of all the predictions based on the positive class. This is also known as positive predictive value and can be depicted by the formula (2) where we have our correct predictions in the numerator for the positive class divided by all the predictions for the positive class including the false positives.

$$Precision = \frac{TP}{TP\ +\ FP}. \quad (2)$$

Recall is defined as the number of instances of the positive class that were correctly predicted. This is also known as hit rate, coverage, or sensitivity and can be depicted by the formula (3) where we have our correct predictions for the positive class in the numerator divided by correct and missed instances for the positive class, giving us the hit rate.

$$Recall = \frac{TP}{TP\ +\ FN}. \quad (3)$$

F1 score is another accuracy measure that is computed by taking the harmonic mean of the precision and recall and can be represented by the formula (4).

$$F1\,Score = \frac{2\ *\ Precision\ *\ Recall}{Precision\ +\ Recall}. \quad (4)$$

3.3. Main methods and techniques

This research will rely on NLP techniques since the primary piece of information in the dataset is raw text. NLP or Natural Language Processing is a part of computer science, human language, and artificial intelligence whose goal is to make a program capable of “understanding” the content of documents, including the contextual nuances of the language within them.

The first step is the normalization of data. Data normalization is a process consisting of steps that should be followed to wrangle, clean, and standardize textual data into a form that machine learning models could consume.

Text normalization steps:

1. Tokenization. It is the process of splitting or segmenting text from sentences into their constituent words.
2. Removing special symbols such as punctuation or emojis.
3. Expanding contractions such as “won’t” or “can’t”.
4. Case conversion. Transforming all tokens to lowercase or uppercase.
5. Removing stop words, words that have little or no significance. They are removed to retain words having maximum significance and context.
6. Stemming. It is the process of reducing a word to its stem that affixes to suffixes and prefixes or the roots of words.
7. Lemmatization.

The next step after the normalization of the text is its vectorization. Text vectorization is the process of converting text into a numerical representation. It is done since machine learning models can not understand the text as is and require the data's numeric representation. This research will use two popular vectorization methods: bag of words and normalized TF-IDF.

The bag of words model is one of the most straightforward yet powerful techniques to extract features from text documents. The essence of this model is to convert text documents into vectors such that each document is converted into a vector representing the frequency of all the distinct words present in the document vector space for that specific document.

TF-IDF stands for Term Frequency-Inverse Document Frequency, a combination of two metrics: term frequency and inverse document frequency. This technique was initially developed as a metric for ranking functions for showing search engine results based on user queries and has come to be a part of information retrieval and text feature extraction now.

The cleaned and vectorized data is forwarded as input into the model for training. This research will focus on decision trees, SVM, multinomial naive bayes, and logistic regression models to find out which of them, within specific conditions, give the best results. These models were selected since they are well-established and reliable. In addition, they are capable of drawing a line between different features in a multi-dimensional space detecting the optimal line between the trolling and non-trolling tweets.

4. Experiment

In order to identify the best model that is preferred to be used for detecting trolling content, we will run multiple experiments using various machine-learning models that will be tested under different conditions.

For the experiments, the CoLab Research Google platform will be used. The programming language is Python since it is well-supported in CoLab. In addition, the programming language has many libraries for analyzing data and training models. All machine learning models, such as logistic regression or multinomial naive bayes and vectorization packages, will be taken from the sklearn Python package. The pandas library is required for experiments as it provides data reading and manipulation functions. The pyplot library will be used for data visualization, and nltk will be used for text preprocessing.

The experiment consists of 3 parts and is visualized in Figure 2.

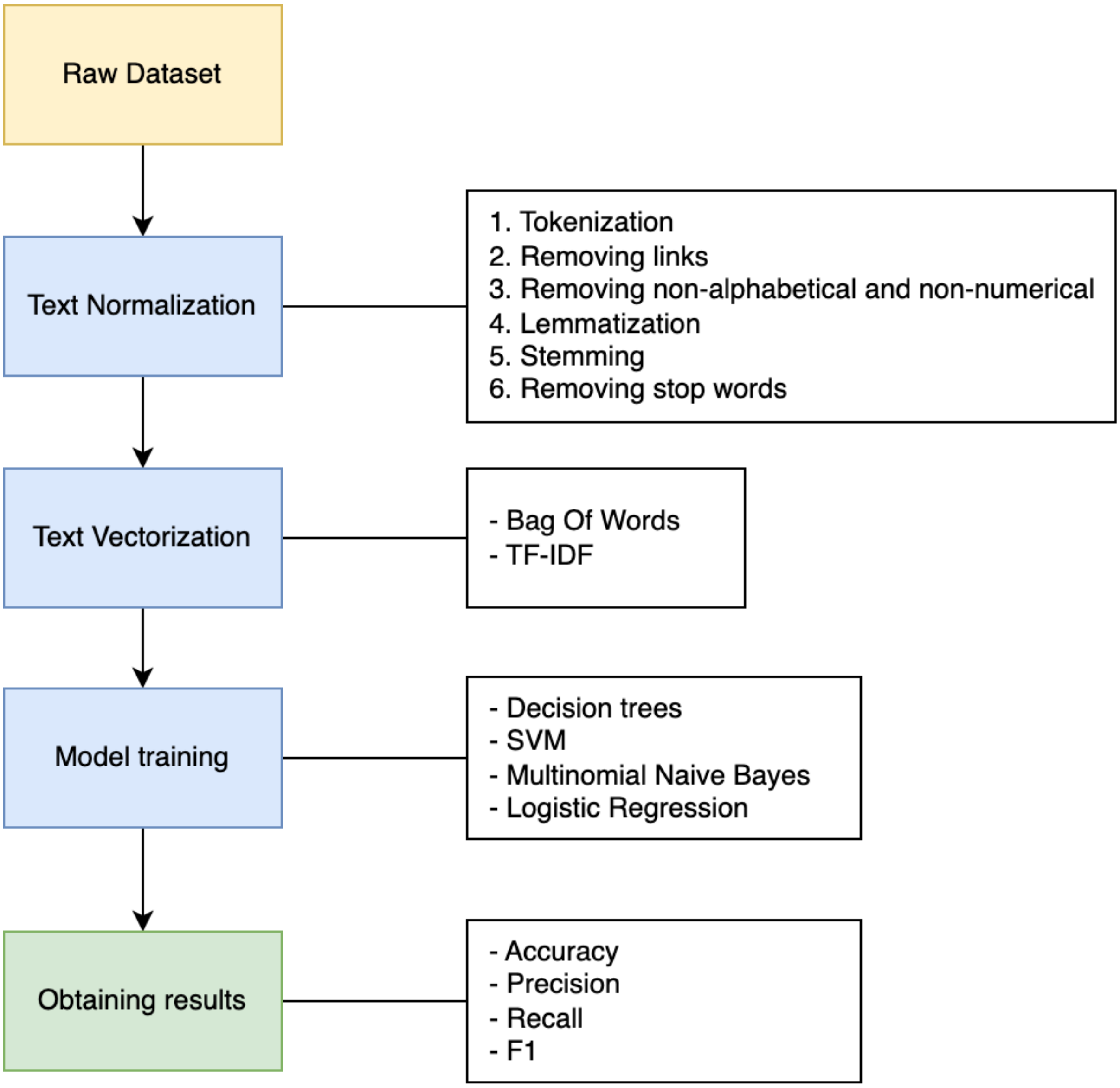


Figure 2: Visualization of the experiment.

The first step in the experiment is data preparation. The entire dataset containing 6000 samples and represented as a CSV file will be read using the "pandas" library. After this, column "TweetId" will be dropped since it does not give any value to models and exists only as a reference to the original tweet for extra information. As a result, the dataset containing data will consist of two columns: "Text" and "IsTrolling". The column "Text" is represented as an arbitrary text that contains links, emojis, and stop words such as articles "the" or "a". The column "IsTrolling" is represented as a number one or zero.

After this, it necessary to perform text normalization. It is required because it can positively affect the results of experiments.

The normalization of text consists of the following steps:

- Words tokenization, a process of splitting a sentence into separate words, which will simplify the performing of the next steps.
- Cleaning all website links using regular expressions. It is required because they do not bring meaningful information and can confuse a machine learning model for experimenting.
- Cleaning everything except alphabetical and numerical characters.
- Lemmatization, a process of grouping the inflected forms of a word so they can be analyzed as a single item.
- Stemming, a process of reducing inflected words to their word stem.
- Removing stop words.

The normalization is done using regular expressions for cleaning up text from needless data and using the “nltk” library. The “nltk” has already implemented functions for lemmatization, stemming, and removing stop words.

The second step is text vectorization. It is necessary because machine learning models can not directly work with text and need data to be represented as numbers, so vectorization is the process of converting text into numerical data. There are various algorithms of vectorization, but this research will use bag of words and normalized TF-IDF since they are the most popular and are available out of the box on the colab.

The final step is the training of models and obtaining results. Vectorized data will be split in the ratio of 75% and 25% to have training and validation sets. When the text is normalized, vectorized, and divided into training and validation chunks, ML models will be trained. For the first experiment, the support vector machine model will be used. The second experiment will use the decision tree model. The multinomial naive bayes will be used for the third one, and the last experiment will use logistic regression.

Every model will have separate experiments for bag of words and normalized TF-IDF vectorization algorithms. In addition, every model will be tested with different levels of text normalization. There will be experiments with:

1. Fully normalized text.
2. Normalization without stemming.
3. Normalization without stemming and lemming.
4. Normalization without stemming, lemming and removing stop words.
5. Normalization without stemming, lemming, removing stop words, and cleaning not alphabetical characters.

After every run of the experiment, the following metrics will be collected: accuracy, recall, precision, and F1, and saved in a separate table for analysis. As a result, it should be possible to determine what model has the better output and should be used for detecting trolling tweets.

5. Results

Consider obtained results of experiments conducted under different conditions using different machine learning models, algorithms of vectorization, and different levels of text normalization. Each chapter demonstrates results for the concrete model, but with different algorithms for text vectorization and different levels of text normalization.

5.1. Support vector machine

Table 2 presents results with complete normalization of text.

Table 2
Results for fully normalized text.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	86.53	86.53	86.55	86.53
Normalized TF-IDF	86.93	86.93	86.96	86.92

Table 3 presents results for text without stemming.

Table 3
Results without stemming.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	85.93	85.93	86.07	85.93
Normalized TF-IDF	87.46	87.46	87.54	87.46

Table 4 presents results for text without stemming and lemming.

Table 4
Results without stemming and lemming.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	86.06	86.06	86.07	86.06
Normalized TF-IDF	88.00	88.00	88.00	88.00

Table 5 presents results for text without stemming, lemming, and removing stop words.

Table 5
Results without stemming, lemming and removing stop words.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	84.46	84.46	84.51	84.46
Normalized TF-IDF	87.80	87.80	88.01	87.78

Table 6 presents results for text without stemming, lemming, removing stop words, and without cleaning everything except alphabetical and numerical characters.

Table 6
Results without stemming, lemming, removing stop words, and cleaning not alphabetical chars.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	84.20	84.20	84.25	84.20
Normalized TF-IDF	88.46	88.46	88.56	88.46

5.2. Decision tree

Table 7 presents results with complete normalization of text.

Table 7
Results for fully normalized text.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	77.13	77.14	77.13	77.12
Normalized TF-IDF	79.13	79.13	79.13	79.13

Table 8 presents results for text without stemming.

Table 8
Results without stemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	80.20	80.40	80.20	80.10
Normalized TF-IDF	77.13	77.14	77.13	77.12

Table 9 presents results for text without stemming and lemming.

Table 9
Results without stemming and lemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	81.06	81.08	81.06	81.06

Normalized TF-IDF	80.00	80.00	80.00	79.99

Table 10 presents results for text without stemming, lemming, and removing stop words.

Table 10
Results without stemming, lemming and removing of stop words.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	75.66	75.68	75.66	75.66
Normalized TF-IDF	79.80	80.06	79.80	79.76

Table 11 presents results for text without stemming, lemming, removing stop words, and without cleaning everything except alphabetical and numerical characters.

Table 11
Results without stemming, lemming, removing of stop words and cleaning not alphabetical and numerical characters.

	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Bag of words	75.46	75.46	75.46	75.46
Normalized TF-IDF	77.53	77.53	77.53	77.51

5.3. Multinomial naive bayes

Table 12 presents results with complete normalization of text.

Table 12
Results for fully normalized text.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	89.20	89.25	89.20	89.19
Normalized TF-IDF	87.13	87.47	87.13	87.10

Table 13 presents results for text without stemming.

Table 13
Results without stemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	88.00	87.99	88.00	87.99
Normalized TF-IDF	88.80	88.87	88.80	88.79

Table 14 presents results for text without stemming and lemming.

Table 14
Results without stemming and lemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	89.40	89.40	89.40	89.39
Normalized TF-IDF	87.80	87.80	87.80	87.80

Table 15 presents results for text without stemming, lemming, and removing stop words.

Table 15
Results without stemming, lemming and removing of stop words.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	88.06	88.09	88.06	88.06
Normalized TF-IDF	87.66	87.66	87.66	87.66

Table 16 presents results for text without stemming, lemming, removing stop words, and without cleaning everything except alphabetical and numerical characters.

Table 16
Results without stemming, lemming, removing stop words, and cleaning not alphabetical chars.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	87.73	87.73	87.73	87.73
Normalized TF-IDF	88.86	88.87	88.86	88.86

5.4. Logistic regression

Table 17 presents results with full normalization of text.

Table 17
Results for fully normalized text.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	85.80	85.80	85.80	85.79
Normalized TF-IDF	85.46	85.47	85.47	85.46

Table 18 presents results for text without stemming.

Table 18
Results without stemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	86.00	86.00	86.00	85.99
Normalized TF-IDF	85.13	85.18	85.13	85.13

Table 19 presents results for text without stemming and lemming.

Table 19
Results without stemming and lemming.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	87.33	87.34	87.33	87.33
Normalized TF-IDF	86.60	86.60	86.60	86.60

Table 20 presents results for text without stemming, lemming and removing of stop words.

Table 20
Results without stemming, lemming and removing of stop words.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	88.53	88.53	88.53	88.53
Normalized TF-IDF	84.06	84.08	84.06	84.06

Table 21 presents results for text without stemming, lemming, removing of stop words and without cleaning everything except alphabetical and numerical characters.

Table 21
Results without stemming, lemming, removing of stop words and cleaning not alphabetical and numerical characters.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Bag of words	87.26	87.37	87.26	87.26
Normalized TF-IDF	84.60	84.60	84.60	84.60

6. Discussion

After analyzing the results (metrics) listed above, it is possible to conclude that the multinomial naive bayes model provides the best result with the bag of words algorithm for text vectorization. The result of multinomial naive bayes is 89.4% in all metrics. At the same time, the worst results demonstrated the decision tree with 75.46% in all metrics using the bag of words vectorizer. The difference between the multinomial naive bayes and the decision tree is 13.94%. The worst result for the multinomial naive bayes was 87.13% with normalized TF-IDF, while the best outcome for the decision tree was 81.06% with the bag of words algorithm.

Logistic regression demonstrated 88.53% accuracy, which is second among the tested models. The logistic regression achieved this result with the bag of words vectorization algorithm and the text that was not normalized. The difference between the best and the worst result for this model is 4.47%.

The SVM model took third place and achieved 88.46%, which is only 0.07% lower than the logistic regression. This result was achieved using the normalized TF-IDF algorithm with the raw text that was not normalized. The difference between this model's best and worst results is 4%. Figure 3 visualizes discussed results and shows the difference between the best and the worst result for every model.

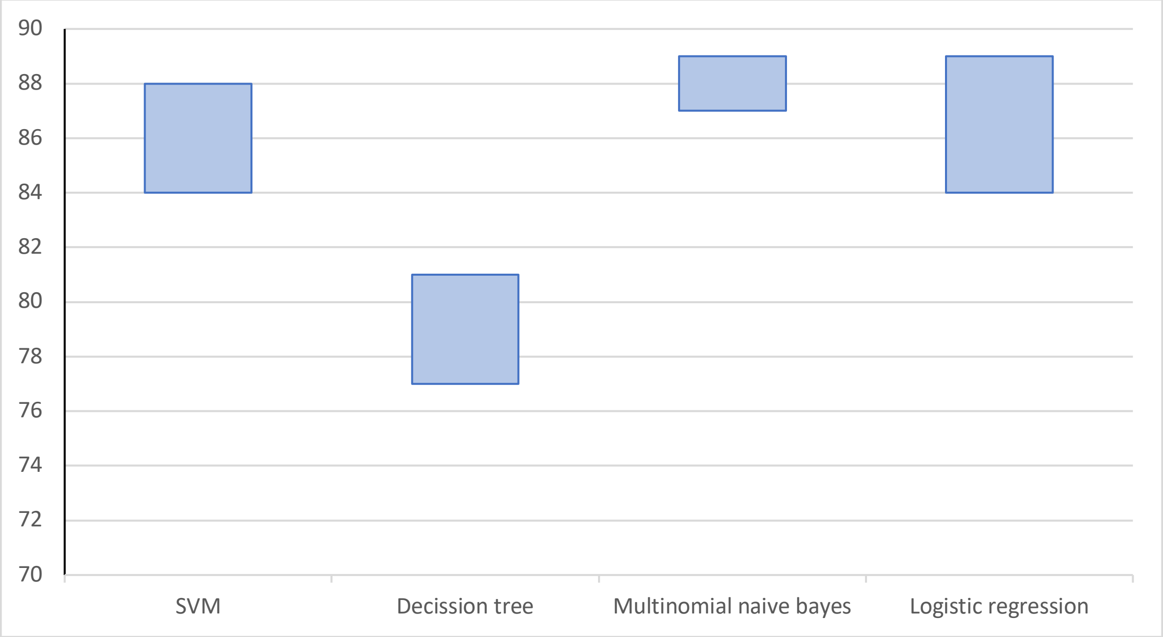


Figure 3: Visualization of obtained results.

An interesting fact is that only sometimes the text normalization leads to improvement of results. Although there is no direct correlation in metrics when text normalization is done or not, overall, this practice may positively affect results. For instance, the SVM model demonstrates good results with the bag of words vectorizer and full-text normalization. However, the best result among all experiments this model gave was when the text was not normalized, and the model used the normalized TF-IDF vectorizer. Proper text normalization can improve the model's performance by 4.26%.

There is no clear performance correlation in vectorizer algorithms since different models with various levels of text normalization demonstrated different results. However, normalized TF-IDF gave better results in 3 out of 4 models with the text without normalization. The most significant difference between vectorization algorithms is 4.47%, so it is worth trying different vectorization approaches to improve metrics.

Based on the analyzed results, it is possible to recommend using the multinomial naive bayes model for detecting trolling content since it demonstrated the best result among all models. In addition, using the normalized TF-IDF vectorizer is preferred because it will most likely demonstrate a better result. Also, it is not recommended to use fully normalized text, and it is better to try different normalization levels to find that normalization level that will improve metrics.

7. Conclusion

The researchers gathered 6000 tweets during this research, where every tweet was labeled as anti-Ukrainian or pro-Ukrainian. The researchers selected four popular machine learning models and conducted experiments to identify which ML model is the most suitable for identifying trolling content in Tweets. The data samples were split in a 75% and 25% ratio for training and validating models. Google colab was used as the experiment environment. This platform allows utilizing a programming language called Python, which is popular in ML and Data Science and has an enormous number of libraries for machine learning.

Every model was tested under different conditions. They were tested with different algorithms of text vectorization and with different levels of text normalization. Unexpectedly text normalization only sometimes improves the performance metrics of models. For instance, the SVM model demonstrated better performance results with no normalized text among all experiments conducted for the model.

The multinomial naive bayes showed the best results for the selected tweets with completely normalized text and the bag of words vectorization algorithm. At the same time, the worst results were obtained from the decision tree in combination with the bag of words vectorization algorithm and without text normalization.

The results of the current research possibly could be used in big data methods for E-learning systems trying to optimize the learning process for teachers and students, which were described in the work [38]. The primary idea of such systems is organizing information stored in libraries with unstructured data from emerging outlets such as social media.

8. Reference

[1] A. Ochoa et al., "Analysis of Cyber-bullying in a virtual social networking," 2011 11th International Conference on Hybrid Intelligent Systems (HIS), Melacca, Malaysia, 2011, pp. 229-234, doi: 10.1109/HIS.2011.6122110.

[2] M. Behzadi, I. G. Harris and A. Derakhshan, "Rapid Cyber-bullying detection method using Compact BERT Models," 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2021, pp. 199-202, doi: 10.1109/ICSC50631.2021.00042.

[3] L. Canales, C. Strapparava, E. Boldrini and P. Martínez-Barco, "Intensional Learning to Efficiently Build Up Automatically Annotated Emotion Corpora," in IEEE Transactions on Affective Computing, vol. 11, no. 2, pp. 335-347, 1 April-June 2020, doi: 10.1109/TAFFC.2017.2764470.

[4] K. B. Raj, J. K. Seth, K. Gulati, S. Choubey, I. Patni and Bhawna, "Automated Cyberstalking Classification using Social Media," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-6, doi: 10.1109/ICSES55317.2022.9914337.

[5] S. Alhazbi, "Behavior-Based Machine Learning Approaches to Identify State-Sponsored Trolls on Twitter," in IEEE Access, vol. 8, pp. 195132-195141, 2020, doi: 10.1109/ACCESS.2020.3033666.

[6] L. Gao, Y. Wu, X. Xiong and J. Tang, "Discriminating Topical Influencers Based on the User Relative Emotion," in IEEE Access, vol. 7, pp. 100120-100130, 2019, doi: 10.1109/ACCESS.2019.2929548.

[7] L. Recalde, J. Mendieta, L. Boratto, L. Terán, C. Vaca and G. Baquerizo, "Who You Should Not Follow: Extracting Word Embeddings from Tweets to Identify Groups of Interest and Hijackers in Demonstrations," in IEEE Transactions on Emerging Topics in Computing, vol. 7, no. 2, pp. 206-217, 1 April-June 2019, doi: 10.1109/TETC.2017.2669404.

[8] Luca Follis; Adam Fish, "3 When to Hack," in Hacker States , MIT Press, 2020, pp.73-111.

[9] Kate Eichhorn, "5 JOURNALISM AND POLITICS AFTER CONTENT," in Content , MIT Press, 2022, pp.103-127.

[10] H. S. Dutta and T. Chakraborty, "Blackmarket-Driven Collusion Among Retweeters–Analysis, Detection, and Characterization," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1935-1944, 2020, doi: 10.1109/TIFS.2019.2953331.

[11] Ross Anderson, "Electronic and Information Warfare," in Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 2020, pp.777-814, doi: 10.1002/9781119644682.ch23.

[12] H. Berghel, "Trolling Pathologies," in Computer, vol. 51, no. 3, pp. 66-69, March 2018, doi: 10.1109/MC.2018.1731067.

[13] H. Berghel and D. Berleant, "The Online Trolling Ecosystem," in Computer, vol. 51, no. 8, pp. 44-51, August 2018, doi: 10.1109/MC.2018.3191256.

[14] N. Colnerič and J. Demšar, "Emotion Recognition on Twitter: Comparative Study and Training a Unison Model," in IEEE Transactions on Affective Computing, vol. 11, no. 3, pp. 433-446, 1 July-Sept. 2020, doi: 10.1109/TAFFC.2018.2807817.

[15] N. Colnerič and J. Demšar, "Emotion Recognition on Twitter: Comparative Study and Training a Unison Model," in IEEE Transactions on Affective Computing, vol. 11, no. 3, pp. 433-446, 1 July-Sept. 2020, doi: 10.1109/TAFFC.2018.2807817.

[16] T. Zhong, T. Wang, J. Wang, J. Wu and F. Zhou, "Multiple-Aspect Attentional Graph Neural Networks for Online Social Network User Localization," in IEEE Access, vol. 8, pp. 95223-95234, 2020, doi: 10.1109/ACCESS.2020.2993876.

[17] K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrechko and O. Ponomarenko, "Efficiency of image convolution," 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019, pp. 578-583, doi: 10.1109/CAOL46282.2019.9019450.

[18] K. Smelyakov, A. Chupryna, O. Bohomolov and I. Ruban, "The Neural Network Technologies Effectiveness for Face Detection," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 201-205, doi: 10.1109/DSMP47368.2020.9204049.

[19] K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko, "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

[20] F. Yao, X. Sun, H. Yu, W. Zhang, W. Liang and K. Fu, "Mimicking the Brain's Cognition of Sarcasm From Multidisciplines for Twitter Sarcasm Detection," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 1, pp. 228-242, Jan. 2023, doi: 10.1109/TNNLS.2021.3093416.

[21] M. Xu, X. Zhang and L. Guo, "Jointly Detecting and Extracting Social Events From Twitter Using Gated BiLSTM-CRF," in IEEE Access, vol. 7, pp. 148462-148471, 2019, doi: 10.1109/ACCESS.2019.2947027.

[22] L. Belcastro, R. Cantini, F. Marozzo, D. Talia and P. Trunfio, "Learning Political Polarization on Social Media Using Neural Networks," in IEEE Access, vol. 8, pp. 47177-47187, 2020, doi: 10.1109/ACCESS.2020.2978950.

[23] Z. Luo, Q. Li and J. Zheng, "Deep Feature Fusion for Rumor Detection on Twitter," in IEEE Access, vol. 9, pp. 126065-126074, 2021, doi: 10.1109/ACCESS.2021.3111790.

[24] A. Nguyen, A. Longa, M. Luca, J. Kaul and G. Lopez, "Emotion Analysis Using Multilayered Networks for Graphical Representation of Tweets," in IEEE Access, vol. 10, pp. 99467-99478, 2022, doi: 10.1109/ACCESS.2022.3207161.

[25] I. Shubin, I. Kyrychenko, P. Goncharov and S. Snisar, "Formal representation of knowledge for infocommunication computerized training systems," 2017 4th International Scientific-Practical

Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, 2017, pp. 287-291, doi: 10.1109/INFOCOMMST.2017.8246399.

[26] Krivoulya G., Tokariev V., Ilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022). CEUR Workshop Proceedings., 12-13 may. 2022 y. - Gliwice, Poland, 2022. - P. 1531-1541.

[27] Clare Llewellyn, Laura Cram, Adrian Favero, and Robin L. Hill. Russian Troll Hunting in a Brexit Twitter Archive. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL '18). Association for Computing Machinery, New York, NY, USA, 361–362. 2018. URL: https://doi.org/10.1145/3197026.3203876/.

[28] A. Ramalingaiah, S. Hussaini, S. Chaudhari. Twitter bot detection using supervised machine learning. ICMAI 2021. Journal of Physics: Conference Series. 1950 (2021) 012006. 2021. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1950/1/012006/pdf/.

[29] Bilal Ghanem, Davide Buscaldi, and Paolo Rosso. TexTrolls: Identifying Russian Trolls on Twitter from a Textual Perspective. 2019. URL: https://arxiv.org/pdf/1910.01340.pdf/.

[30] Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, Xinshun Feng, Qingyue Zhang, Hongrui Wang, Yuhan Liu, Yuyang Bai, Heng Wang, Zijian Cai, Yanbo Wang, Lijing Zheng, Zihan Ma, Jundong Li, Minnan Luo. 2023. TwiBot-22: Towards Graph-Based Twitter Bot Detection. 36th Conference on Neural Information Processing Systems, 12 Feb 2023. URL: https://arxiv.org/pdf/2206.04564.pdf/.

[31] C. W. Seah, H. L. Chieu, K. M. A. Chai, L. Teow, and L. W. Yeong. Troll detection by domain-adapting sentiment analysis. In 2015 18th International Conference on Information Fusion (Fusion). 2015. pp. 792–799.

[32] Paolo Fornacciari, Monica Mordonini, Agostino Poggi, Laura Sani, and Michele Tomaiuolo. A holistic system for troll detection on Twitter. Computers in Human Behavior 89 (2018), 258 – 268. 2018. URL: https://doi.org/10.1016/j.chb.2018.08.008/.

[33] C.J. Hutto and Eric Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media. 2015. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14550/14399/.

[34] Jose Lorenzo C. Capistrano, Jessie James P. Suarez, and Prospero C. Naval. SALSA: Detection of Cybertrolls Using Sentiment, Aggression, Lexical and Syntactic Analysis of Tweets. In Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics (WIMS2019). Association for Computing Machinery, New York, NY, USA, Article Article 10, 6 pages. 2019. URL: https://doi.org/10.1145/3326467.3326471/.

[35] DataTurks. Tweets Dataset for Detection of Cyber-Trolls, 2020. URL: https://www.kaggle.com/datatūrks/dataset-for-detection-of-cybertrolls/.

[36] Todor Mihaylov and Preslav Nakov. Hunting for Troll Comments in News Community Forums, 2019. URL: https://arxiv.org/pdf/1911.08113.pdf/.

[37] FiveThirtyEight. Tweets Dataset for Russian-Troll, 2017. URL: https://www.kaggle.com/datasets/fivethirtyeight/russian-troll-tweets/.

[38] Sharonova, N., Kyrychenko, I., Tereshchenko, G., "Application of big data methods in E-learning systems", 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 1302-1311.