electronics

MDPI

Article

A Multilayered Preprocessing Approach for Recognition and Classification of Malicious Social Network Messages

Aušra Čepulionytė, Jevgenijus Toldinas * and Borisas Lozinskis

Department of Computer Science, Kaunas University of Technology, 44249 Kaunas, Lithuania; ausra.cepulionyte@ktu.edu (A.Č.); borisas.lozinskis@ktu.lt (B.L.)
* Correspondence: eugenijus.toldinas@ktu.lt

Abstract: The primary methods of communication in the modern world are social networks, which are rife with harmful messages that can injure both psychologically and financially. Most websites do not offer services that automatically delete or send malicious communications back to the sender for correction, or notify the sender of inaccuracies in the content of the messages. The deployment of such systems could make use of techniques for identifying and categorizing harmful messages. This paper suggests a novel multilayered preprocessing approach for the recognition and classification of malicious social network messages to limit negative impact, resulting in fewer toxic messages, scams, and aggressive comments in social media messages and commenting areas. As a result, less technical knowledge would be required to investigate the effects of harmful messages. The dataset was created using the regional Lithuanian language with four classes: aggressive, insulting, toxic, and malicious. Three machine learning algorithms were examined, five use cases of a multilayered preprocessing approach were suggested, and experiments were conducted to identify and classify harmful messages in the Lithuanian language.

Keywords: method for classification of social network messages; recognition of malicious messages; preprocessing method; machine learning

Citation: Čepulionytė, A.; Toldinas, J.; Lozinskis, B. A Multilayered Preprocessing Approach for Recognition and Classification of Malicious Social Network Messages. *Electronics* **2023**, *12*, 3785. https://doi.org/10.3390/electronics12183785

Academic Editors: Umit Karabiyik, Abdelkader Ouda and Mamoun Alazab

Received: 22 August 2023
Revised: 5 September 2023
Accepted: 6 September 2023
Published: 7 September 2023

1. Introduction

The Internet is increasingly essential for interacting with others and exchanging information as a single platform in which users leave digital records of their own behavior and usage patterns, which, if properly analyzed, might provide crucial information about user behavior [1,2]. Tweets, blogs, chat messaging, and other forms of social media are the primary means of community communication today, and Internet-based crime and can be used offline to investigate crimes or, in real time, to prevent them [3]. According to Stacy Jo Dixon, in the first quarter of 2023, Facebook had approximately three billion active monthly users [4]. The amount of financial harm caused by cybercrime that was reported to the Internet Crime Complaint Center (IC3) rose considerably between 2001 and 2022. The annual financial loss as a result of complaints referred to the IC3 increased from USD 6.9 billion to USD 10.3 billion in the most recent reported period [5].

Social media platforms must understand the fundamentals of human social interaction and find simple, effective ways to maintain the necessary standards of confidentiality, security, and reliability. To use and manipulate the vast amount of information on the social web, governments, intelligence agencies, and technical specialists must step forward and try to adopt new technologies and paradigms [6]. Sentiment analysis is a way of determining a text's sentiment polarity, which is used to identify whether the text is conveying a positive or negative message [7]. To automatically determine the sentiment polarity of a comment is the aim of the sentiment classification of online social networks (OSNs). It requires investigation into handling emotional ranges to achieve a better interpretation of OSN messages, because messages can have a range of sentiments in addition to positive or negative ones, including neutral and neutral with gradations [8].

Machine learning approaches make it easier to develop models from sample data, speeding up decision-making processes based on real-world inputs. These techniques allow learning from input data via descriptive statistics as well as production values within a predetermined range [9]. Input data from a batch or the real-time collection of data instances are needed for machine learning algorithms to train their models. The terms "data point," "vector," "event," "sample," "case," "object," "record," "observation," and "entity" can all be used to describe a single datum instance [10]. Unlabeled data are utilized in unsupervised learning since it lacks additional information while labeled data have useful tags and are used in supervised learning. Benchmark datasets are used in machine learning for model accuracy comparisons and performance measures.

The main contributions of this paper are:

- A novel categorized dataset of malicious social media messages in the Lithuanian language is created with four classes of messages: aggressive, insulting, toxic, and malicious;
- A multilayered preprocessing approach of Lithuanian social media messages and a detailed experimental analysis to decide which machine learning algorithm would produce the best results for the classifier;
- The empirical quantification of message class recognition for the proposed approach, allowing us to achieve accuracy from 95% to 98% in classifying malicious social network messages.

The remaining parts of the paper are organized as follows. Section 2 discusses the related works. Section 3 presents and explains the methodology. Section 4 presents and discusses the experimental results. Finally, Section 5 presents the conclusions.

2. Related Works

Based on data that can be found on social networks, the information is separated into four categories: hyperlinks, images, audio, and text (a subset of spoken language primarily produced with a text or string to examine the content) [11]. OSNs are receiving attention from users who are malicious or abnormal and engage in malicious activities such as harassing others, plotting attacks (in which terrorists may be involved), and disseminating false information [12]. Spam is the term for unsolicited messages that are sent in large quantities by fostering a sense of community trust. Spammers engage in illegal acts including phishing, advertising, surveillance, assault against women, and cyberbullying, among others [13]. Instead of using legitimate accounts, spammers typically distribute spam using fraudulent, compromised, or cloned accounts, crowd-sourcing strategies, and automated bots [14]. The taxonomy of various social spam detection techniques and approaches are observed as follows: URL list-based spam filtering (Blacklist, Whitelist, Greylist), honeypot and honeynet-based spam detection, and machine learning (ML) and deep learning (DL)-based social spam detection. ML and DL are used for social spam content detection including malicious URL detection [15] and text-based spam detection [16,17].

Social media bots (SMBs) are tools that people and organizations employ to spread information, expand their reach, and boost their impact. Malicious bots can annoy or burden users by participating in unethical actions, including stealing the identities of real users, persuading voters to favor politicians, spreading hate speech, and other divisive material [18]. SMBs are classified into three main groups: benign bots, neutral bots, and malicious bots. For SMB detection, the most used ML methods are random forest (RF), SVM, and AdaBoost, while LSTM and CNN are the two most widely used DL algorithms; unfortunately, there is a lack of large datasets to train models [18].

In [19], bidirectional encoder representations from transformers (BERT) are proposed. It has been shown that the pre-training of linguistic models is effective in improving many tasks related to the processing of natural languages, including the intuition and paraphrase of natural languages, the recognition of named entities and, the answer to questions. The development of pre-trained language models based on transformer architectures has stimulated the evolution of modern techniques for many tasks in the field of natural language

processing (NLP) [20–22]. The study of [23] proposed text classification using BERT for natural language processing and the results of the experiment showed that combinations of BERT with CNN, RNN, and BiLSTM performed well with precision, recall rate, and F1 score, compared to Word2vec. The new BSTC (BERT, SKEP, and TextCNN) fake review detection model is proposed [24] based on a pre-trained language model and a convolutional neural network. The highest accuracy was achieved with all three gold standard datasets (Hotel, Restaurant, and Doctor), with 93.44%, 91.25%, and 92.86%, respectively. The process of choosing, modifying, and transforming raw data into features that can be utilized to enhance the performance of machine learning models is known as feature engineering. In some tasks, effective feature engineering combined with conventional machine learning methods could produce outcomes comparable to BERT [25]. Although there has been a rise in interest in learning general-purpose sentence representations, the majority of the research in that field has been conducted in English and has mostly been monolingual [26].

Spam is typically defined as undesired text that is sent or received over social media platforms like Facebook, Twitter, YouTube, e-mail, etc. [27]. The authors of [28] proposed a novel four-layered, state-of-the-art detection strategy, with graph-based, neighbor-based, automation-based, and time-based features to find spammers on social networking sites. The majority of SMS spam classifiers use supervised algorithms like Naïve Bayes (NB), support vector machine (SVM), neural networks, and regression, because the availability of the output column (labeled data) of the SMS dataset makes it possible to train classification models [29]. Using a total of 20 samples from the dataset (SMS Spam Corpora and Twitter Corpora), the suggested solution in [30] employs reinforcement learning to identify the malicious social bots. It also makes use of k-nearest neighbor (KNN) and a recurrent neural network (RNN). A social bot is a computer program that uses an application programming interface (API) to operate a social media account. It can be used for malicious activities, such internet trolling and fraud. Bots are classified as malicious or benign in the study cited [31].

Information phishing began as a marketing tactic, but it has since evolved into destructive internet interactions that expose users to significant security risks using tools including emails, comments, blogs, and messaging. Given their adaptability and ability to make the most of current hardware and computational limitations, deep learning architectures like convolutional neural networks (CNNs), multi-layer perceptrons (MLPs), and the long short-term memory (LSTM) have been successfully used for email spam classification [32]. The identification of fake news [33,34] is a difficult challenge for social media platforms like Facebook, Twitter, etc., because of the volume of data that people publish on these sites. To determine whether a news article is authentic or fake, a deep CNN for fake news detection was presented in [35] and models were tested using binary class datasets. For NLP researchers, sarcasm presents a formidable challenge and can entirely alter the meaning of a statement, making it challenging for modern models and systems to recognize it. In order to create models that can accurately identify the settings in which sarcasm may occur or is suitable, an approach for the automatic detection of sarcasm context has been developed [36].

Cyber social media security examines the dynamics of online social networks, the data's vulnerabilities, and the potential effects of their abuse by social media attackers. Due to their nature, the volume of content they include, and the sensitive information they use, social media are the most attack-prone section of the internet [37–39]. To classify a social media message as a part of a particular crisis event, it is important to take into account a number of factors, such as the message's nature, the information it contains, the source of that information, its credibility, the timing, and its location [40]. Some of the features can be automatically extracted, whereas some need to be manually labeled. The best performance is achieved with an ensemble approach for the identification and classification of crime-related tweets that uses logistic regression (LR), SVMs, KNN, a decision tree (DT), and an RF classifier assigned the weights of 1, 2, 1, and 1, respectively, ensemble together via a soft weighted voting classifier along with a term frequency–inverse document frequency (TF-

IDF) vectorizer with an accuracy of 96.2% on the testing dataset [41]. When compared to the ground truth labeled by network experts, an RNN-LSTM model that was trained to identify five different social engineering attacks (SEA) that may show signs of information gathering achieves classification precision and recall scores of 0.84 and 0.81, respectively [42].

3. Materials and Methods

In this section, we describe the created dataset and proposed method for the recognition and classification of malicious social network messages in the Lithuanian language.

3.1. Essentials of Lithuanian Words Formation

The use of a representative dataset is required for machine learning to be successful. This is because it is crucial to choose a dataset that accurately reflects the settings for the model's real-world applications while training a machine learning algorithm [43]. The dataset collected for the model to recognize and classify malicious social network messages in the Lithuanian language included benign, aggressive, insulting, toxic, and malicious data. A Lithuanian classified dataset of messages was created for the recognition and classification of malicious messages on social networks. The dataset had to be processed before submitting it to the machine learning algorithm, so it was important to analyze the word formation of the Lithuanian language and know how the messages in the dataset could be processed.

There is currently no BERT-scale monolingual NLP model for Lithuanian. Worldwide, it has comparatively very few speakers. However, Lithuanian is typically included in the majority of pre-trained multilingual models given that it is the native language of one of the European Union's member states [44]. Significant parts of Lithuanian word formation [45] include the following:

- Stem—a part of a word without an ending, for example, "padainavim" in the word "padainavimas";
- The prefix is the part of the stem that precedes the root, for example "pa" in the word "padainavimas";
- The root is the main part of related words, for example, "dain" in the words "daina", "priedainis";
- The suffix is a part of the stem that follows the root, for example "el" in the word "dainelė";
- The ending is the part of the end of the word that changes depending on the meaning and the grammatical relationship with other words, for example "a", "os", "oje" in the words "daina", "dainos", "dainoje";
- Interjection—these are sounds interspersed in the root, as, for example, "n" and "m" in the words "sninga", "limpa";
- Particle—"si", for example, in the word "pasiruošti".

The Lithuanian language is full of short words that do not have a clear meaning when classifying malicious messages, and include the following [45]:

- Conjunctions (ir, kad, bet. . .);
- Coincidences (oh, brr, che. . .);
- Emoticons (o, a, ė, ak. . .);
- Particles (ne, nė, lyg, dar, argi. . .);
- Prepositions (ant, už, po. . .);
- Pronouns (jos, tas, mes, šis. . .).

Word formation in the Lithuanian language is quite complicated for message processing in a Lithuanian dataset. The Lithuanian language is full of short words, which in the machine learning model would be like noise, so when programming text processing, it would be necessary to define which words or which parts of words could be removed from the message. Message processing techniques must be explored through experimental testing.

3.2. Dataset for Recognition and Classification of Malicious Social Network Messages in Lithuanian Language

The term "machine learning" refers to algorithms that analyze training data for patterns and then utilize those patterns to classify, forecast, or perform other tasks without having explicit programming instructions for doing so [46]. The proposed classification system for the Lithuanian dataset has four classes that describe the message content and was designed for use in the training and assessment of ML techniques for harmful message recognition in OSNs:

- **Aggressive** messages are written with a fixed or outright aggressive mood or tone. Such a message may be worded to provoke anger or objection and may even contain threatening or offensive content. Such a message can be used to achieve one's goals, but it can also cause conflicts, worsen relationships, and harm the quality of communication;
- **Insulting** messages are written with an established or directly insulting mood or tone. Such a message may be worded in such a way that it may hurt another person, or his feelings or dignity, and may sometimes even contain threatening content or pose a security threat. An insulting message can cause emotional trauma, create tension and cause conflict between individuals. In addition, such messages can damage interpersonal relationships and create a negative reputation for the sender;
- **Toxic** messages written in such a way that it creates negative or harmful feelings for the recipient and has a negative impact on the relationship between them. Such a message may be worded in such a way as to provoke anger, hatred, insults, or may be intended to repel or criticize the recipient. A toxic message can be very damaging because of its effects on emotional, psychological, and physical health. The message text can cause stress, anxiety, depression, insecurity, and more. In addition, such messages can cause interpersonal conflicts and destroy relationships between individuals. All swear words are classified as toxic messages;
- **Malicious** messages are sent with the intention of causing panic or fear in a person to make a faster decision that will benefit the scammers. Such messages can be worded in such a way as to induce fear and panic in a person to click on a link or reply to the email, which can lead to financial losses or exposed personal information to fraudsters. The messages are often sent with dangerous messages such as "Your bank account has been frozen!", "Your account has been compromised!", "Your data is in dangerous hands!", etc. Such messages are often sent with a request for an immediate response or to click on a link to resolve the issues. The malicious message may send applications (bots) that encrypt the user's data, and then demands a ransom to be paid to restore the data. Often, such messages can be crafted to appear to be from your bank, insurance company, social network, or another trusted source.

A message could match multiple classes, i.e., to be both aggressive and toxic, or insulting, aggressive, and toxic at the same time. The generic dataset creation process is depicted in Figure 1.

OSN text messages were picked and collected in a text file using REST API software. Then experts in the field decided which messages could be classified as aggressive, insulting, toxic, malicious, or could be assigned to multiple classes, and they labeled the messages accordingly. The labels had values of 0 or 1 that characterize the message. A value of 0 means that the message does not belong to that class, and a value of 1 means that it does. An example of the created dataset labeling is presented in Table 1.

Message number zero was labeled as malicious, number one was labeled as toxic, number two was labeled as insulting, and number three and four as aggressive. For an ML classifier to work properly, the dataset should be diverse and balanced, with similar percentages of benign and malicious messages, and so more benign messages may be needed to maintain balance and improve classifier performance. The created Lithuanian benchmark dataset was saved as a .csv file. The distribution of messages is shown in Figure 2.

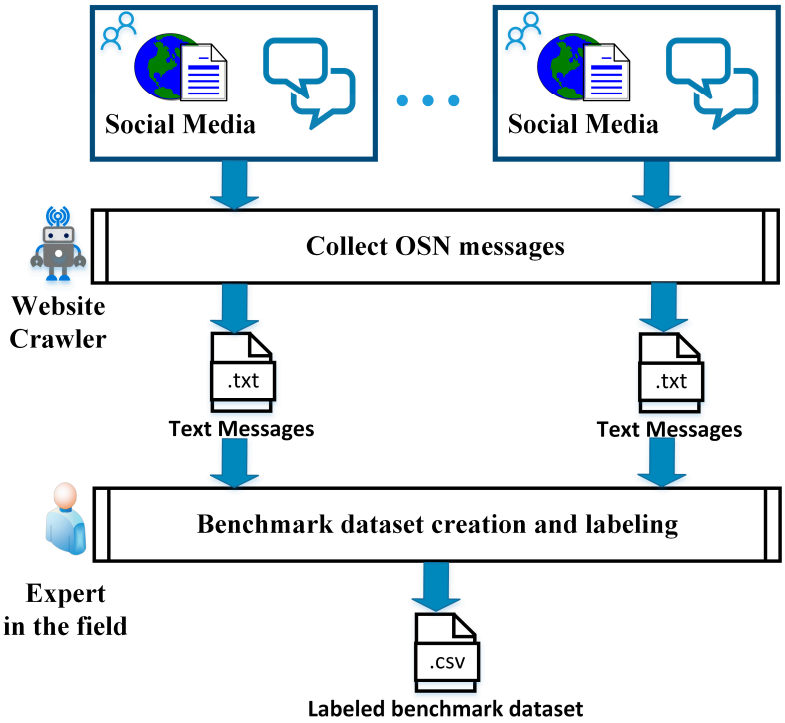
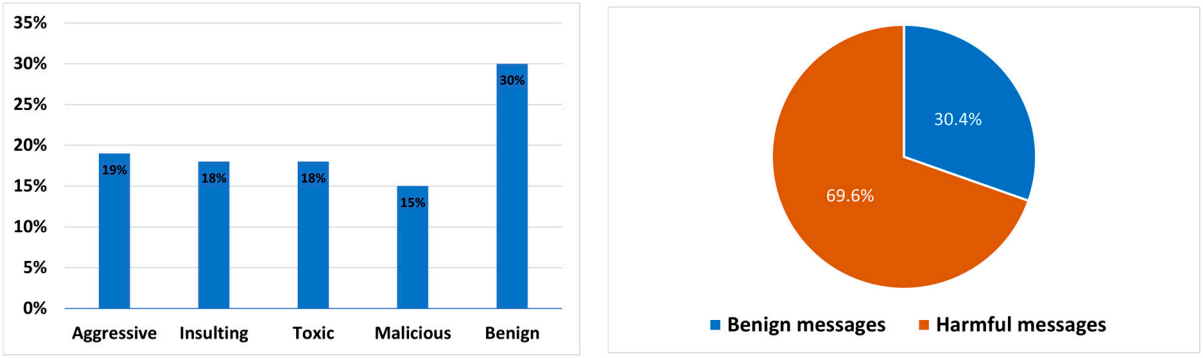


Figure 1. Generic dataset creation process.

Table 1. Dataset messages labeling.

No.	Aggressive	Insulting	Toxic	Malicious	Message Text in Lithuanian and English Translation
0	0	0	0	1	Mirė senelis, kuris kasė deimantus Afrikoje, paveldėjo turtą 1,000,000 $ The grandfather who mined diamonds in Africa died, inherited a fortune of $1,000,000 data
1	0	0	1	0	Po šimts gegučių After a hundred cuckoos
2	0	1	0	0	Žalčio koja pastaroji The snake's leg is the latter
3	1	0	0	0	Suk tave devynios Turn you nine
4	1	0	0	0	Kad tu kiaurai žemę prasmegtum So that you can penetrate the earth



(**a**) (**b**)

Figure 2. Distribution of messages (**a**) based on the categories; (**b**) benign and harmful.

The dataset was carefully prepared based on the specific characteristics of the Lithuanian language and was a collection of messages that represented each class in the Lithuanian language most effectively. In some word compositions, a message may have conveyed a positive sentiment, while in others, it may have been toxic, insulting, or aggressive. As depicted in Figure 2a, our dataset was structured as follows: 30% of the messages were benign and did not belong to any class, 24% were aggressive, 21% were insulting, 25% were toxic, and 15% were malicious messages. To maintain balance, we included more benign messages in the created dataset. Because some messages may have belonged to several harmful classes simultaneously, the total distribution of messages based on the categories (including benign) was 115%. We had over 600 messages in the dataset for the experiment, where 69.6% were harmful and 30.4% were benign, as depicted in Figure 2b. The Lithuanian dataset that was created for the training and testing of the ML models was randomly divided into 80% for the training and 20% for testing.

3.3. Multilayered Preprocessing Approach for Social Media Messages in Lithuanian Language

The text of social media posts is shortened instead of being in complete sentences, and so if less text is present in the data, the performance of models with a high dependence on the text may degrade [47]. Smaller text is often used when blogging on a particular topic, ignoring auxiliary elements such as suffixes and endings, and shortening the word to its stem [48]. To resolve the mentioned issues, we proposed a multilayered preprocessing approach consisting of four layers that could be used sequentially, one-by-one, or independently:

Layer 1. The message's special characters are removed, and the text is changed to lowercase letters, leaving only words;
Layer 2. The Lithuanian characters [ą,č,ę,ė,į,š,ų,ū,ž] are replaced by Latin characters accordingly [a,c,e,e,i,s,u,u,z] in the message;
Layer 3. Endings such as -a, -as, -yje, -us, etc., are removed from each word;
Layer 4. Words shorter than 4 characters are removed.

Vectorization is applied for the pre-processed messages using the CountVectorizer method from the Python programming language library. The parameters of the CountVectorizer method are as follows:

- analyzer = 'word'. The parameter means that the unit of text analysis is a word, rather than some other unit of text, such as a character or part of a sentence bullet;
- binary = False. The parameter specifies whether the vector of words should be binary. When binary = False, the vector of words is numeric and indicates the number of times each word occurs in the text. For example, if the text contains the word "labas" 3 times, the corresponding value in the vector will be 3;
- decode_error = 'strict'. The parameter is used to read text files when there are problems decoding text characters. It specifies how sequences of characters that cannot be decoded into text should be handled;
- min_df = 2. The parameter specifies the minimum number of words in the data at which a word must appear to be considered significant and included in the vector;
- ngram_range = (1, 3). The parameter indicates how many consecutive words will be combined.

3.4. Recognition and Classification Method for Social Media Messages in Lithuanian Language

An illustration of the proposed method for identifying and classifying malicious messages in the Lithuanian language is presented in Figure 3.

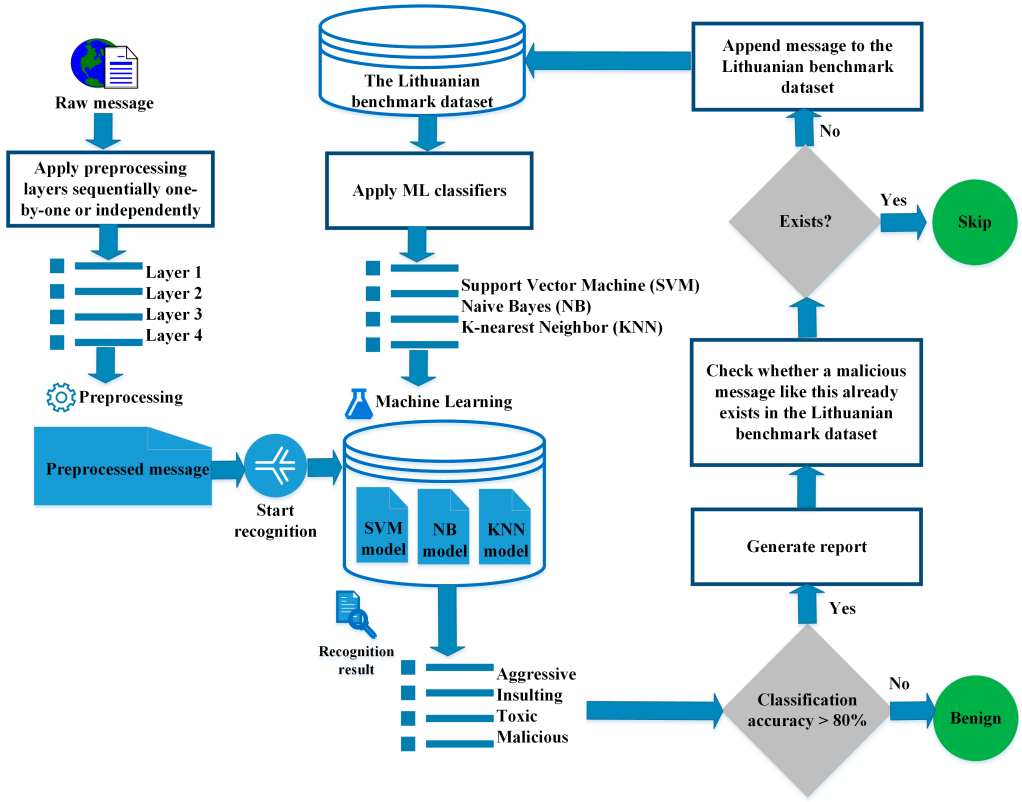


Figure 3. Illustration of the proposed method for identifying and classifying malicious messages in Lithuanian language.

The method uses the newly created labeled Lithuanian benchmark dataset, in which messages are classified into four classes: aggressive, insulting, toxic, and malicious. The Lithuanian language is full of short and unimportant words that might produce unwanted noise in the machine learning classifier and negatively affect the classifier's performance, because messages are frequently written in lowercase and without Lithuanian letters. The solution takes advantage of the suggested multilayered preprocessing approach to prevent misclassification issues. The pre-processing layers can be used sequentially, one-by-one, or independently.

The Naïve Bayes (NB), support vector machines (SVMs), and the k-nearest neighbor (KNN) methods are among the most widely used and successful ML algorithms for classifying texts, according to an analysis of related works. These ML algorithms were specifically utilized when putting into practice the suggested method for recognizing and classifying malicious messages in Lithuanian social networks.

The following steps were taken to classify a message:

1. A vectorized array of words was passed to the ML classifier, which used the predict_proba method of the Python library to predict the probability that the message belonged to the class;
2. Predictions were assessed, and if the message was more than 80% likely to be malicious, a report was generated with the metadata received from the HTML request and the ML classifier prediction results;
3. Following the generation of the report, it was determined whether a malicious message like this already existed in the created Lithuanian benchmark dataset;
4. If the message classified as malicious was missing from the created Lithuanian benchmark dataset, a new labeled message was added to the Lithuanian benchmark dataset based on the classifier's findings.

4. Experimental Settings and Results

The architecture of the system for the proposed method of evaluation is depicted in Figure 4.

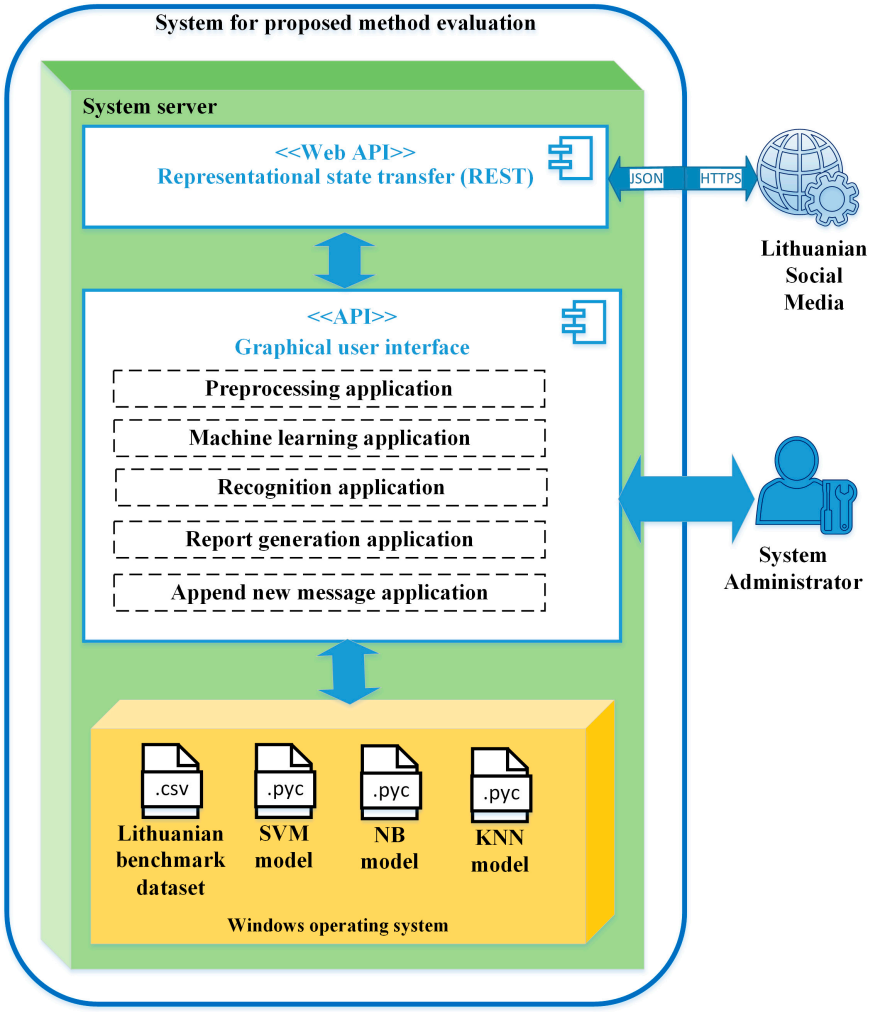


Figure 4. Architecture of the system for proposed method evaluation.

The main blocks of the system are:

- The labeled Lithuanian benchmark dataset in .csv file;
- The trained models for the Lithuanian benchmark dataset using SVM, NB, and KNN in .pyc files;
- The graphical user interface that includes tools for pre-processing messages obtained from the web, training and evaluating ML models, adding new malicious messages to the Lithuanian benchmark dataset, and generating reports;
- The interface was also designed to graphically display obtained results.

The report generated by the system includes classifier results and metadata received from the HTML request:

- ID—unique request number;
- Host—the domain from which the message was sent;
- Remote_Address—IP address;
- Timestamp—the time of the message when the sender wrote the message;
- Message—a message written by the sender;
- Results—results of the classifier;
- Metadata—HTML request metadata.

4.1. Experimental Settings

The prototype of the proposed system architecture was created using the following technologies:

1. The Microsoft Visual Studio Code tool for creating software code due to the convenient use of plugins and writing software code;
2. The Python programming language, due to the fast processing of large amounts of data.
3. Open source libraries in the Python programming language:
    - Pandas, a NumPy library with libraries for data analysis and table manipulation;
    - Scikit-learn, which has machine learning-implemented algorithm libraries;
    - The tornado web framework and asynchronous network library suitable for web applications and large amounts of data processing;
    - The BeautifulSoup library for processing HTML and XML documents and analyzing web pages.
4. Machine learning algorithms:
    - Support vector machine (SVM);
    - Multinomial Naïve Bayes (NB);
    - K-nearest neighbor (KNN).
5. Dataset in the Lithuanian language;
6. Using the REST API technology (GET and POST methods) for obtaining data from the social network and displaying the data.
7. Laptop for prototype development and prototype experimental testing:
    - Operating system: Microsoft Windows 11;
    - Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz, 2803 Mhz, 4 Core(s), 8 Logical processors;
    - Physical memory (RAM): 16.0 GB.

4.2. Experimental Results Evaluating Proposed Lithuanian Benchmark Dataset

The proposed Lithuanian benchmark dataset could be decomposed into training and testing data. The train_test_split method of the scikit-learn library was used to split the data into training and testing datasets, which gave the processed messages the value of x and classes with attributes the value of y, specified as a 20% test set build parameter and random generator initial state 2 (train_test_split (x, y, test_size = 0.20, random_state = 2)).

The experiments were conducted using ML algorithms, including SVM, multinomial Naïve Bayes (NB), and k-nearest neighbor (KNN):

- For SVM, the scikit-learn library SVC was used with an average setting of the balance between the error forgiveness and edge fitting, a linear function, a polynomial function with a degree of 3, an automatic gamma parameter, and a probability for each class (SVC (C = 1.0, kernel = 'linear', degree = 3, gamma = 'auto', probability = True));
- In the case of NB, the scikit-learn library MultinomialNB was used with the smoothing parameter 1 and the calculation set in frequency classes (MultinomialNB (alpha = 1.0, fit_prior = True));
- In the case of KNN, the scikit-learn library KNeighborsClassifier was used with a set parameter of 13 neighbors (KNeighborsClassifier (n_neighbors = 13)).

To determine the classification accuracy of the model, the accuracy_score method of the scikit-learn library was used, which was provided with a test set with classes and features y_test, as well as predictions of the ML method (accuracy_score (y_test, classifier_prediction)). Using the mean method of the pandas library and performing additional mathematical steps, i.e., by dividing the number of guessed messages in a class by the total number of messages in the class, the accuracy of the classifier was calculated.

Each algorithm was evaluated and compared with other researchers' obtained results using an accuracy that is defined as follows:

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN), \quad (1)$$

where, True Positive (TP)—how many cases of "positive" are correctly predicted; True Negative (TN)—how many cases of "negative" are correctly predicted; False Positive (FP)—how many cases of "positive" are incorrectly predicted; False Negative (FN)—how many cases of "negative" are incorrectly predicted.

The experimental results are presented in Table 2.

Table 2. Experimental results evaluating proposed Lithuanian benchmark dataset.

Research	Language of the Dataset	Classes	Accuracy		
			SVM	NB	KNN
Abbas et al., 2019 [49]	English	Sentiment positive Sentiment negative	-	0.8992	-
Asogwa et al., 2022 [50]	English	Offensive Non-offensive	0.99	0.50	-
Toktarova et al., 2022 [51]	English	Hate speech Other	0.92	0.89	0.90
Poojitha et al., 2023 [52]	English	Toxic Non-toxic	0.94	-	0.86
Fouad et al., 2022 [53]	Arabic	Fake news Non-fake news	0.859	0.823	0.806
Proposed	Lithuanian	Malicious Benign	0.85	0.86	0.74

As we can see from Table 2, state-of-the-art ML technologies and the best feature-based algorithms have been applied to recognize and classify texts in the English language. In [54], the authors conducted some preliminary experiments on the classification of sexism with five datasets: two for English, one for Spanish, one for Italian, and one for Portuguese, with multilingual BERT. The experiments show a low generalization between multilingual datasets. The best result was achieved by generalizing between the two English datasets. This indicates that a multilingual generalization approach probably performs worse than an intralingual approach [54].

The accuracies of the ML algorithms by class are presented in Table 3.

Table 3. The accuracies of the ML algorithms by class.

ML Algorithm	Accuracy			
	Aggressive	Insulting	Toxic	Malicious
NB	0.82	0.84	0.80	0.97
KNN	0.55	0.83	0.77	0.79
SVM	0.83	0.85	0.79	0.94

The multinomial Naïve Bayes (NB) method is the most suitable for classifying Lithuanian messages (see Table 2) since its accuracy is higher than SVM accuracy and higher than k-nearest neighbor (KNN) accuracy.

4.3. Experimental Results Applying Preprocessing Layers for Lithuanian OSN Messages

A social media message received from the OSN has requests in the HTML format, from which only the text of the message is retrieved. Prior to classification, the message is preprocessed: special characters are removed, and the text is changed to lowercase letters, leaving only words (**Layer 1**); Lithuanian letters are changed to Latin letters (Layer 2);

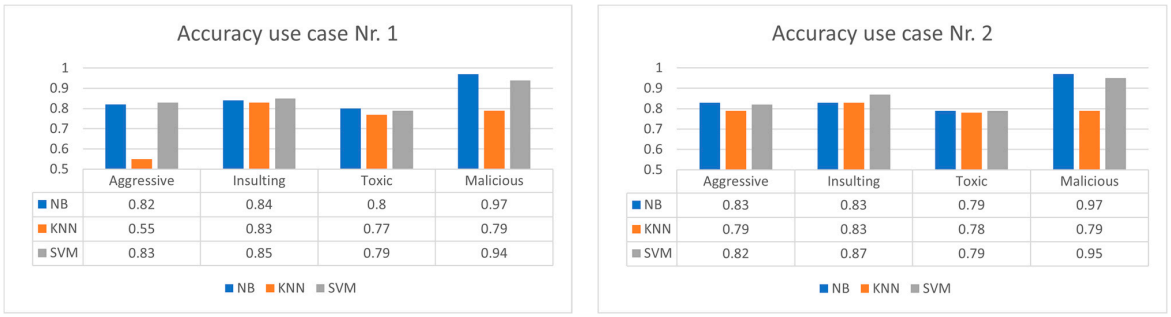
endings are removed from each word (Layer 3); words shorter than four characters are removed (Layer 4).

After preprocessing, the message is vectorized. A vectorized message's attributes for the message itself are defined by the attributes 0 (the word does not belong to the message) and 1 (the word belongs to the message). The vectorized message is sent to a trained ML classifier to calculate the probabilities for each class, who determines whether the message is benign or not. To evaluate the applicability of the proposed preprocessing layers, we decided to conduct experiments using five scenarios that are presented in Table 4.

Table 4. Experimental scenarios applying proposed preprocessing layers approach.

Use Case	Layer 1. The Message's Special Characters Are Removed, and the Text Is Changed to Lowercase Letters, Leaving Only Words.	Layer 2. The Lithuanian Characters [ą,č,ę,ė,į,š,ų,ū,ž] Are Replaced by Latin Characters Accordingly [a,c,e,e,i,s,u,u,z] in the Message	Layer 3. Endings Such as -a, -as, -yje, -us, etc., Are Removed from Each Word	Layer 4. Words Shorter than Four Characters Are Removed
1.	Applicable	Applicable	Applicable	Applicable
2.	Applicable	Not applicable	Applicable	Shorter than three
3.	Applicable	Applicable	Not applicable	Shorter than three
4.	Applicable	Applicable	Applicable	Shorter than three
5.	Applicable	Not applicable	Not applicable	Shorter than three

For each message text preprocessing scenario, an evaluation of the ML models was performed to calculate the probability of the message matching to a class. Prediction accuracy according to class was calculated by how accurately the classifier identified messages correctly according to each class separately using Equation (1). The experimental results for all scenario use cases are presented in Figures 5–7.



(**a**) (**b**)

Figure 5. The experimental results for scenario use cases: (**a**) preprocessing use case Nr. 1; (**b**) preprocessing use case Nr. 2.

(**a**) (**b**)

Figure 6. The experimental results for scenario use cases: (**a**) preprocessing use case Nr. 3; (**b**) preprocessing use case Nr. 4.

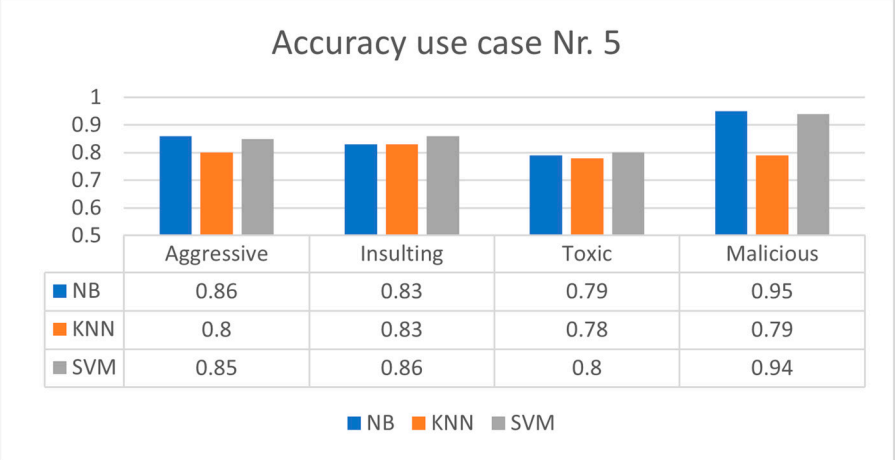


Figure 7. The experimental results for scenario preprocessing use case No. 5.

The NB method achieves the best accuracy in the first scenario use case, whereas the NB and SVM methods achieve the highest accuracy in the second scenario use case.

In the third scenario use case, the SVM method achieves the best accuracy (1–3%), outperforming NB for aggressive, insulting, and toxic messages. In the fourth scenario use case, the SVM method similarly achieves the best accuracy (1–5%), outperforming NB, and only for malicious messages the NB method 3% outperforms the SVM method.

The NB and SVM methods perform nearly similarly in the fifth scenario use case but show the best accuracy across all classes. For aggressive message recognition, the fifth scenario use case performs the best in terms of classification accuracy. The recognition of an insulting message works best in the second scenario use case. The ability to recognize toxic messages is best in the third scenario use case. The optimum scenario use case for identifying malicious messages is the fourth one.

Since the structure of the dataset consists of four classes, the average accuracy classification score for all classes is determined by the formula

$$\text{Average accuracy} = (TP\_a + TP\_b + TP\_c + TP\_d)/(n\_a + n\_b + n\_c + n\_d), \quad (2)$$

where:

- TP_a—number of cases of the “Aggressive” class correctly predicted;
- TP_b—number of cases of the “Insulting” class correctly predicted;
- TP_c—number of cases of the “Toxic” class correctly predicted;
- TP_d—number of cases of the “Malicious” class correctly predicted;
- n_a—total number of the “Aggressive” class members;
- n_b—total number of the “Insulting” class members;
- n_c—total number of the “Toxic” class of members;
- n_d—Total number of the “ Malicious” members.

This study’s findings are shown in Table 5.

Table 5. The average accuracy classification score for all use cases.

ML Method	Average Accuracy Classification Score (Calculated Using Equation (2))				
	Use Case No. 1	Use Case No. 2	Use Case No. 3	Use Case No. 4	Use Case No. 5
NB	0.86	0.86	0.85	0.85	0.86
KNN	0.74	0.8	0.81	0.78	0.8
SVM	0.85	0.86	0.87	0.86	0.86

A graphical representation of the average accuracy classification scores is depicted in Figure 8.

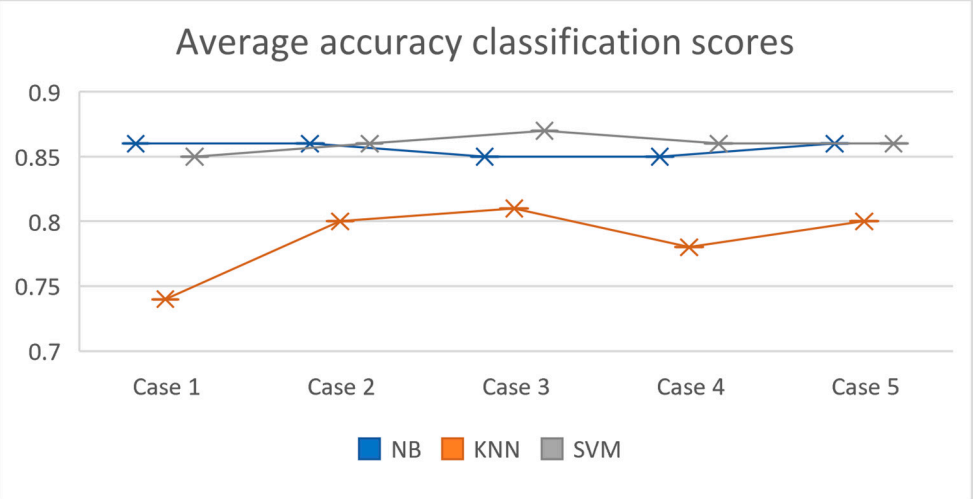


Figure 8. Graphical representation of the average accuracy classification scores.

An average accuracy of 86% shows that both the NB and the SVM methods perform fairly well. The SVM method in the third case (no word ending removal and applied elimination of phrases shorter than three characters) performed the best in comparison with all other scenarios and methods. The third case also shows the best performance of the KNN method, and even though the accuracy of the NB method in this situation was the lowest compared to the previous ones, it was still only 1%.

5. Conclusions

Machine learning (ML) is one of the most popular technologies available today, enabling systems to learn and improve through experience without explicitly coding them. ML methods are widely used in NLP. The Naïve Bayes (NB), support vector machine (SVM), and k-nearest neighbor (KNN) methods are among the most widely used and successful ML algorithms for classifying texts. State-of-the-art ML technologies and the best feature-based algorithms have been applied to recognize and classify texts in the English language.

This study offers a unique multilayered preprocessing approach for recognizing and classifying malicious OSN messages in the Lithuanian language. Five use cases of the proposed multilayered preprocessing approach were evaluated. Our experimental results indicate that according to the scenarios, the NB and SVC methods work similarly in all use cases, and their average accuracy is about 86%. The SVC method in the third use case (which does not apply word ending removal and applies the removal of words shorter than three characters) performed the best compared to all other scenarios and classifiers, which leads to the conclusion that the third scenario is the most suitable for classifying harmful messages in Lithuanian using the SVC method. The NB method demonstrates only 2% less accuracy for the third use case.

Evaluating the performance of the ML methods used according to the five use cases of the proposed multilayered preprocessing approach, it was found that KNN is the most unsuitable for recognizing and classifying harmful OSN messages in Lithuanian. Furthermore, by comparing the results of our experiment with state-of-the-art results (see Table 2), we were able to recognize and classify harmful messages using the NB method with 86% accuracy.

The particularly low accuracy rate of KNN in the "Aggressive" class can be attributed to the characteristics of our dataset. KNN relies on the presence of repeated words or patterns to make predictions effectively. In this specific dataset, we have a limited number of repeated messages containing the same words that represent the "Aggressive" class. Since KNN depends on identifying similarities between data points based on these repeated patterns, its performance is hindered when there is a scarcity of such repeated instances. Consequently, the algorithm struggles to accurately classify messages in the "Aggressive"

category due to the lack of sufficient repeated words to make reliable predictions. Based on the results of our research, KNN requires two things: more repeatable data, especially for the "Aggressive" class, and a lower accuracy (not 80%, but within the range of 50–60%). However, it is worth noting that an excessive influx of data can also have a negative impact on model performance.

We did not simply translate the string and use one of the many already existing approaches in the English language because of language specificities. The Lithuanian language has unique linguistic and cultural nuances, which the English language does not. These nuances can have a significant impact on how messages are composed and interpreted. Elements such as humor, sarcasm, or regional slang may not translate effectively, potentially leading to the misclassification or misinterpretation of social network messages.

Our dataset consists of real-world data from Lithuanian social networks, which is valuable for analyzing the specific context and social dynamics within Lithuania. Translating the dataset into English would result in the loss of this contextual richness. Based on the experimental scenarios we used, our research aims to address the global issue of malicious social network messages and provide a solution that can be adapted to multiple languages without the need for translation. Furthermore, translating a large dataset from one language to another can be a resource-intensive task.

We encourage researchers to create datasets using regional languages as a feature direction because doing so may improve the recognition and classification of harmful messages and may be crucial in developing a unified dataset that combines multiple datasets of regional languages and suggests appropriate preprocessing techniques.

Author Contributions: Conceptualization, A.Č., J.T. and B.L.; methodology, J.T.; software, A.Č.; validation, A.Č., J.T. and B.L.; formal analysis and investigation, J.T. and B.L.; resources, A.Č.; data curation, A.Č.; writing—original draft preparation, A.Č., J.T. and B.L.; writing—review and editing, A.Č., J.T. and B.L.; visualization, J.T. and B.L.; supervision, J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is unavailable due to ethical restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Luna, S.; Pennock, M.J. Social media applications and emergency management: A literature review and research agenda. *Int. J. Disaster Risk Reduct.* **2018**, *28*, 565–577. [CrossRef]
2. Bhattacharjee, S.D.; Tolone, W.J.; Paranjape, V.S. Identifying malicious social media contents using multi-view Context-Aware active learning. *Future Gener. Comput. Syst.* **2019**, *100*, 365–379. [CrossRef]
3. Soomro, T.R.; Hussain, M. Social Media-Related Cybercrimes and Techniques for Their Prevention. *Appl. Comput. Syst.* **2019**, *24*, 9–17. [CrossRef]
4. Dixon, S. Social Media-Statistics & Facts. Available online: https://www.statista.com/topics/1164/social-networks/#topicOverview (accessed on 20 July 2023).
5. Statista. Cyber Crime: Reported Damage to the IC3 2022. Available online: https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us (accessed on 20 July 2023).
6. Thakur, K.; Hayajneh, T.; Tseng, J. Cyber Security in Social Media: Challenges and the Way Forward. *IT Prof.* **2019**, *21*, 41–49. [CrossRef]
7. Wanda, P.; Huang, J. Model of Sentiment Analysis with Deep Learning in Social Network Environment. In Proceedings of the 2nd International Conference on Electronic Information and Communication Technology (ICEICT), Harbin, China, 20–22 January 2019. [CrossRef]
8. Wanda, P.; Jie, H.J. DeepSentiment: Finding Malicious Sentiment in Online Social Network based on Dynamic Deep Learning. *IAENG Int. J. Comput. Sci.* **2019**, *46*, 616–627.
9. Mishra, S.; Shukla, P.; Agarwal, R. Analyzing Machine Learning Enabled Fake News Detection Techniques for Diversified Datasets. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1575365. [CrossRef]
10. Toshniwal, A.; Mahesh, K.; Jayashree, R. Overview of Anomaly Detection techniques in Machine Learning. In Proceedings of the Fourth International Conference on I-SMAC, Palladam, India, 7–9 October 2022. [CrossRef]

11. Kondamudi, M.R.; Sahoo, S.R.; Chouhan, L.; Yadav, N. A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101571. [CrossRef]
12. Sharma, K.; Singh, A. A Systematic Review: Detection of Anomalies in Social Networks. In Proceedings of the International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 23–25 March 2023. [CrossRef]
13. Koggalahewa, D.; Xu, Y.; Foo, E. An unsupervised method for social network spammer detection based on user information interests. *J. Big Data* **2022**, *9*, 7. [CrossRef]
14. Rao, S.; Verma, A.K.; Bhatia, T. A review on social spam detection: Challenges, open issues, and future directions. *Expert Syst. Appl.* **2021**, *186*, 115742. [CrossRef]
15. Al-Haija, Q.A.; Al-Fayoumi, M. An intelligent identification and classification system for malicious uniform resource locators (URLs). *Neural Comput. Appl.* **2023**, *35*, 16995–17011. [CrossRef]
16. Martinez-Romo, J.; Araujo, L. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Syst. Appl.* **2013**, *40*, 2992–3000. [CrossRef]
17. Almutlaq, R.; Hafez, A. Detection Mechanism for Malicious Messages on KSU Student Social Network. *Int. J. Data Sci. Technol.* **2020**, *6*, 23–36. [CrossRef]
18. Ellaky, Z.; Benabbou, F.; Ouahabi, S. Systematic Literature Review of Social Media Bots Detection Systems. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101551. [CrossRef]
19. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805. [CrossRef]
20. Pattanaik, B.; Mandal, S.; Tripathy, R.M. A survey on rumor detection and prevention in social media using deep learning. *Knowl. Inf. Syst.* **2023**, *65*, 3839–3880. [CrossRef]
21. Zhang, X.; Malkov, Y.; Florez, O.; Serim Park, S.; McWilliams, B.; Han, J.; El-Kishky, A. TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations. *arXiv* **2022**, arXiv:2209.07562. [CrossRef]
22. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:1901.02860. [CrossRef]
23. Bello, A.; Ng, S.-C.; Leung, M.-F. A BERT Framework to Sentiment Analysis of Tweets. *Sensors* **2023**, *23*, 506. [CrossRef] [PubMed]
24. Lu, J.; Zhan, X.; Liu, G.; Zhan, X.; Deng, X. BSTC: A Fake Review Detection Model Based on a Pre-Trained Language Model and Convolutional Neural Network. *Electronics* **2023**, *12*, 2165. [CrossRef]
25. Gani, R.; Chalaguine, L. Feature Engineering vs BERT on Twitter Data. *arXiv* **2022**, arXiv:2210.16168. [CrossRef]
26. Lample, G.; Conneau, A. Cross-lingual Language Model Pretraining. *arXiv* **2019**, arXiv:1901.07291. [CrossRef]
27. Kaddoura, S.; Chandrasekaran, G.; Popescu, D.E.; Duraisamy, J.H. A systematic literature review on spam content detection and classification. *PeerJ Comput. Sci.* **2022**, *8*, e830. [CrossRef] [PubMed]
28. Bankar, S.H.; Shinde, S.A. Spammer Detection of Social Networking Sites Using 4 Novel Techniques. Available online: https://www.academia.edu/download/34105340/Sachin_Bankar.pdf (accessed on 20 July 2023).
29. Odera, D.; Odiaga, G. A comparative analysis of recurrent neural network and support vector machine for binary classification of spam short message service. *World J. Adv. Eng. Technol. Sci.* **2023**, *9*, 127–152. [CrossRef]
30. Kumar, R.M.; Bharathi, P.S. Detection of Malicious Social Bots with reinforcement learning technique with URL Features in Twitter Network with KNN in comparison with RNN. In Proceedings of the Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 6–7 April 2023. [CrossRef]
31. Mbona, I.; Eloff, J.H.P. Classifying social media bots as malicious or benign using semi-supervised machine learning. *J. Cybersecur.* **2023**, *9*, tyac015. [CrossRef]
32. Baccouche, A.; Ahmed, S.; Sierra-Sosa, D.; Elmaghraby, A. Malicious Text Identification: Deep Learning from Public Comments and Emails. *Information* **2020**, *11*, 312. [CrossRef]
33. Alkhodair, S.A.; Ding, S.H.H.; Fung, B.C.M.; Liu, J. Detecting breaking news rumors of emerging topics in social media. *Inf. Process. Manag.* **2020**, *57*, 102018. [CrossRef]
34. Meel, P.; Vishwakarma, D.K. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Syst. Appl.* **2020**, *153*, 112986. [CrossRef]
35. Kaliyar, R.H.; Goswami, A.; Narang, P.; Sinha, S. FNDNet—A deep convolutional neural network for fake news detection. *Cogn. Syst. Res.* **2020**, *61*, 32–44. [CrossRef]
36. Băroiu, A.-C.; Trăușan-Matu, Ș. Comparison of Deep Learning Models for Automatic Detection of Sarcasm Context on the MUStARD Dataset. *Electronics* **2023**, *12*, 666. [CrossRef]
37. Sharma, S.; Jain, A. Role of sentiment analysis in social media security and analytics. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, 5. [CrossRef]
38. Lippmann, R.P.; Campbell, W.M.; Weller-Fahy, D.J.; Mensch, A.C.; Zeno, G.M.; Campbell, J.P. Finding malicious cyber discussions in social media. *Linc. Lab. J.* **2016**, *22*, 46–59. Available online: https://apps.dtic.mil/sti/citations/AD1034416 (accessed on 3 August 2023).
39. Rahman, M.S.; Halder, S.; Uddin, M.A.; Acharjee, U.K. An efficient hybrid system for anomaly detection in social networks. *Cybersecurity* **2021**, *4*, 10. [CrossRef]

40. Krishna, Y.V.; Jahnavi, G.; Tharun, M.; Yegineti, S.G.; Raja, G.; Suneetha, B. Survey: Analysis of Security Issues on Social Media using Data Science techniques. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 26–28 April 2023. [CrossRef]
41. Siddiqui, T.; Hina, S.; Asif, R.; Ahmed, S.; Ahmed, M. An ensemble approach for the identification and classification of crime tweets in the English language. *Comput. Sci. Inf. Technol.* **2023**, *4*, 149–159. [CrossRef]
42. Aun, Y.; Gan, M.; Wahab, N.H.B.A.; Guan, G.H. Social engineering attack classifications on social media using deep learning. *Comput. Mater. Contin.* **2023**, *74*, 4917–4931. [CrossRef]
43. Damaševičius, R.; Venčkauskas, A.; Toldinas, J.; Grigaliūnas, Š. Ensemble-Based Classification Using Neural Networks and Machine Learning Models for Windows PE Malware Detection. *Electronics* **2021**, *10*, 485. [CrossRef]
44. Stankevičius, L.; Lukoševičius, M. Testing pre-trained Transformer models for Lithuanian news clustering. *arXiv* **2020**, arXiv:2004.03461. [CrossRef]
45. Kalbos Pažinimas: Lietuvių Kalbos Žodžių Daryba, Kaityba, Sandara (Morfologija). Available online: https://lietuviu5-6.mkp.emokykla.lt/lt/mo/zinynas/kalbos_pazinimas_lietuviu_kalbos_zodziu_daryba_kaityba_sandara_morfologija/ (accessed on 3 August 2023).
46. Boyd, K.L. Datasheets for Datasets help ML Engineers Notice and Understand Ethical Issues in Training Data. *Proc. ACM Hum. -Comput. Interact.* **2021**, *5*, 1–27. [CrossRef]
47. Song, J.; Han, K.; Kim, S.-W. "I Have No Text in My Post": Using Visual Hints to Model User Emotions in Social Media. In Proceedings of the ACM Web Conference, Lyon, France, 25–29 April 2022. [CrossRef]
48. Barkovska, O.; Rusnak, P.; Tkachov, V.; Muzyka, T. Impact of Stemming on Efficiency of Messages Likelihood Definition in Telegram Newsfeeds. In Proceedings of the 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 3–7 October 2022. [CrossRef]
49. Abbas, M.; Memon, K.A.; Jamali, A.A.; Memon, S.; Ahmed, A. Multinomial Naive Bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 62–67.
50. Asogwa, D.C.; Chukwuneke, C.I.; Ngene, C.C.; Anigbogu, G.N. Hate Speech Classification Using SVM and Naive BAYES. *IOSR J. Mob. Comput. Appl. (IOSR-JMCA)* **2022**, *9*, 27–34. [CrossRef]
51. Toktarova, A.; Iztaev, Z.; Kozhabekova, P.; Suieuova, N.; Opondo, R.O.; Kerimbekov, M.; Zhunisbekova, Z. Automated Hate Speech Classification using Emotion Analysis in Social Media User Generated Texts. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 6621–6634.
52. Poojitha, K.; Charish, A.S.; Reddy, M.A.K.; Ayyasamy, S. Classification of social media Toxic comments using Machine learning models. *Comput. Sci. Mach. Learn.* **2023**. [CrossRef]
53. Fouad, K.M.; Sabbeh, S.F.; Medhat, W. Arabic fake news detection using deep learning. *Comput. Mater. Contin.* **2022**, *71*, 3647–3665. [CrossRef]
54. Fortuna, P.; Soler-Company, J.; Wanner, L. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Inf. Process. Manag.* **2021**, *58*, 102524. [CrossRef]