Methods of information systems protection

UDC 004. 032.26, 528.854 **doi:** https://doi.org/10.20998/2522-9052.2023.3.09

Andrii Podorozhniak[1], Nataliia Liubchenko[1], Vasyl Oliinyk[1], Viktoriia Roh[2]

[1] National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
[2] Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

RESEARCH APPLICATION OF THE SPAM FILTERING AND SPAMMER DETECTION ALGORITHMS ON SOCIAL MEDIA AND MESSENGERS

Abstract. In the current era, numerous social networks and messaging platforms have become integral parts of our lives, particularly in relation to work activities, due to the prevailing COVID-19 pandemic and russian war in Ukraine. Amidst this backdrop, the issue of spam and spammers has become more pertinent than ever, with a continuous rise in the incidence of spam within work-related text streams. Spam refers to textual content that is extraneous to a specific text stream, while a spammer denotes an individual who disseminates unsolicited messages for personal gain. The proposed article is devoted to address this scientific and practical challenge of identifying spammers and detecting spam messages within the textual context of any social network or messenger. This endeavor encompasses the utilization of diverse spam detection algorithms and approaches for spammer identification. Four algorithms were implemented, namely a naive Bayesian classifier, Support-vector machine, multilayer perceptron neural network, and convolutional neural network. The research objective was to develop a spam detection algorithm that can be seamlessly integrated into a messenger platform, exemplified by the utilization of Telegram as a case study. The designed algorithm discerns spam based on the contextual characteristics of a specific text stream, subsequently removing the spam message and blocking the spammer-user until authorized by one of the application administrators.

Keywords: spam; social network; naive Bayesian classifier; Support-vector machine; multilayer perceptron neural network; convolution neural network; spammers detection.

Introduction

Most likely, only email inboxes are equipped with built-in anti-spam algorithms, while other chat rooms lack such functionality. This could explain why the proportion of spam in mailboxes and other messaging platforms is generally similar.

For instance, a malicious link injected into a message and sent to an employee within a company can pose a significant threat to the entire organization. Consequently, the modern world faces the challenge of monitoring incoming text streams in social networks and messengers [1].

It is imperative to detect and prohibit spammers, as this simplifies the operation of algorithms and complicates the efforts of spammers, ultimately reducing the overall prevalence of spam. Ability to filter spam messages, identify spammers, and enforce bans within messengers and social networks holds the potential to save considerable time for humanity and prevent the loss of valuable information and financial resources. To address this problem, we employed algorithms such as the naive Bayesian classifier, support vector method, multilayer perceptron neural network, and convolutional neural network. Additionally, we developed a straightforward algorithm that identifies and blocks users recognized as spammers. By integrating these investigated algorithms, we can begin to tackle the issue of spam within social networks and messengers.

Object, subject and methods of research. The objective of this study is to explore the feasibility of employing various algorithms in the development of software aimed at filtering spam within the textual content of social network messengers. The primary goals are to swiftly respond to spam messages and accurately identify spammers. The study aims to accomplish the following tasks:

a) analyze the specific capabilities of recognizing spam messages;

b) evaluate the existing methods of spam detection;

c) implement spam-fighting methods based on the naive Bayesian Classifier, reference vectors, and multilayer perceptron neural network;

d) analyze the utilized algorithms;

e) examine the fundamental existing algorithms for spam detection;

f) develop and implement spammer detection mechanisms.

The research focuses on the process of identifying spam within the textual context of social network messengers.

The subject of the study revolves around the process of filtering spam in social network messengers by leveraging a range of methods for recognizing spam, as well as identifying and prohibiting spammers.

The research methodology involves employing classification theories, probabilistic classifiers, neural network theory, statistical analysis methods, linguistic techniques, and spammer detection approaches. The study's scientific novelty lies in the enhancement of spam recognition methods within messengers, utilizing the textual content of specific text streams.

Additionally, it encompasses the identification of spammers and the prompt response to messages originating from spammers.

Literature analysis

Spam refers to the mass distribution of unsolicited advertising correspondence to individuals who have not expressed a desire to receive it [2].

To mitigate spam issue, anti-spam filters are utilized to save time. However, these filters can occasionally misclassify important messages as spam, leading to their accidental deletion.

The most effective method of combating spam is to prevent spammers from obtaining one's email address [3].

Auto-Spam Detection Software, commonly known as Anti-Spam Filters, can be employed by end-users or on servers. Such software operates through two primary approaches [4]:

- message content analysis: This algorithm-based approach assesses the message content to determine its spam status. If classified as spam, the message can be marked, moved to a separate folder, or deleted. This software can function on both the server and the client computer. However, with this approach, the spam messages are still received, incurring the associated costs, as the anti-spam software determines whether to display them;
- sender classification: This approach categorizes the sender as a spammer without analyzing the message content. It can only be implemented at the server that directly receives the messages. This method reduces costs by refusing to accept messages from known spammers and contacting other servers for verification. However, the benefits are not as significant as expected since spammers often attempt to bypass such protection measures, necessitating individual handling of each attempt and increasing server overhead.

This project focuses on a statistical Bayesian spam filtering method that incorporates a support vector method and a multilayer perceptron neural network.

Naive Bayesian classifier. The naive Bayes classifier is the simplest of these models, in that it assumes that all attributes of the examples are independent of each other given the context of the class [5]. This is the so-called “naive Bayes assumption”. While this assumption is clearly false in most real-world tasks, naive Bayes often performs classification very well. Mathematically Bayes' theorem is [6, 7]:

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

where $P(A)$ – the probability of A occurring; $P(B)$ – the probability of B occurring; $P(A/B)$ – the probability of A given B; $P(B/A)$ – the probability of B given A.

Support-vector machine. For a given set of training samples, each marked as appropriate to one or the other of two categories, the Support-vector machine (SVM) training algorithm builds a model that assigns new samples to one or the other category, making it a probabilistic binary linear classifier [8, 9], as shown in Fig. 1.

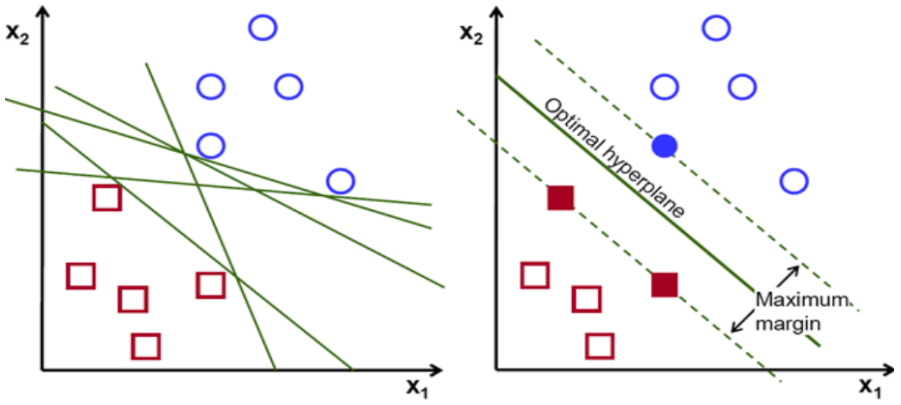

Fig. 1. SVM deals with linearly separate data

Perceptron. The process begins by taking all the input values and multiplying them by their weights. Then, all of these multiplied values are added together to create the weighted sum [10, 11]. The weighted sum is then applied to the activation function, producing the perceptron's output. The activation function plays the integral role of ensuring the output is mapped between required values such as (0,1) or (-1,1). It is important to note that the weight of an input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down [12]. Logic diagram of the basic perceptron is shown in Fig. 2.

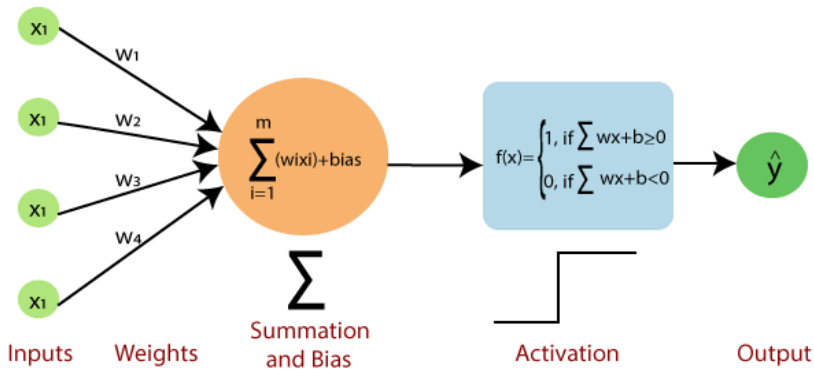

Fig. 2. Logic diagram of the basic perceptron

Convolution neural network. Convolution neural network (CNN) is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks [13], such as convolution layers, pooling layers, and fully connected layers.

The structure of the CNN we used [14, 15] is shown in Fig. 3.

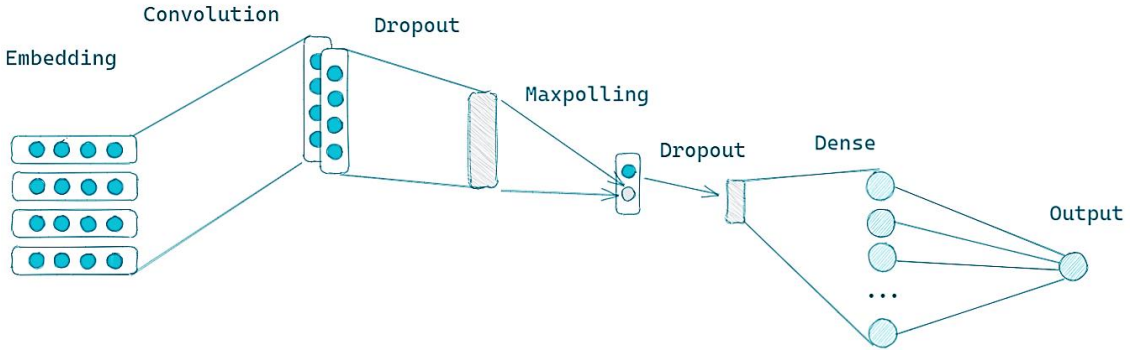

Fig. 3. The structure of the used CNN

Metrics evaluation

Also, in addition to the usual accuracy metric for evaluating selected algorithms, we used F1 score [16].

Accuracy is a ratio between the correctly classified samples to the total number of samples. Nowadays it is the most used metric of classification performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP – (True Positive) correctly classified positive sample; TN – (True Negative) the sample is negative and it is classified as negative; FP – (False Positive) the sample is negative but it is classified as positive; FN – (False Negative) the sample is positive but it is classified as negative. The explanation of the accuracy evaluation is shown in Fig. 4.

Implementation

As a training dataset was chosen the dataset of spam messages from the Kaggle SMS Spam Collection Dataset, but the dataset of messages from a particular company can also be used to train the algorithm [17]. To implement the spam filtering algorithms, we used the Python 3.6 programming language, the PyCharm. programming environment and the Keras, NumPy, Sklearn and Pandas libraries, MySQL DB for storing spammers and all users of the text stream [18, 19].

The simulation was performed on a LifeBook E744 notebook with 8Gb RAM, an Intel Core i7 CPU (up to 3.2 GHz) and an Intel HD Graphics 4600.

	Predicted Positives	Predicted Negatives
Positives	True Positives	False Negatives
Negatives	False Positives	True Negatives

Fig. 4. The explanation of accuracy evaluation

We employed four widely recognized spam recognition algorithms: Naïve Bayesian Classifier, Perceptron, Convolutional Neural Network, and Support Vector Machine. The results of the tests are shown at the Table 1. The spam message analyzing process is shown in Fig. 5.

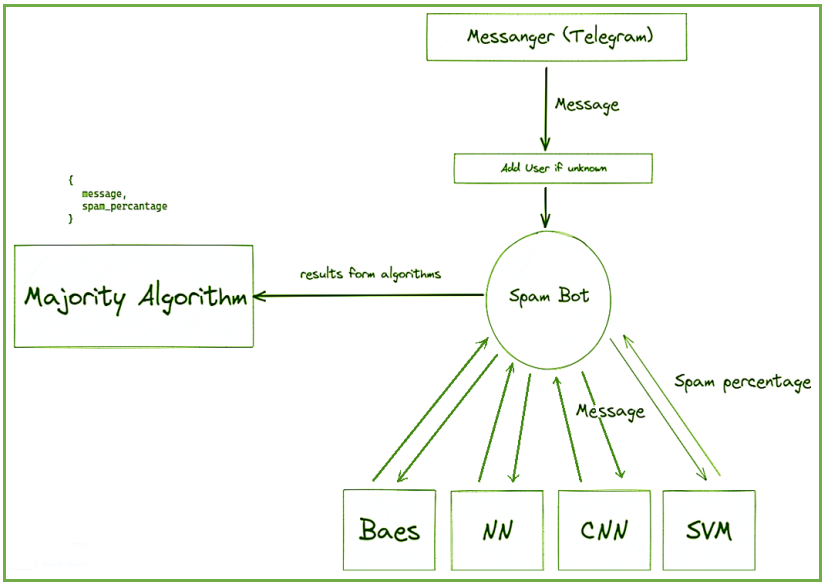

Fig. 5. Spam message analyzing process

Table 1 – **Precision, recall and F1 score of the model**

Algorithm	Training sample	Test sample
Bayes	0.988	0.982
SVM	0.998	0.989
NN	0.997	0.979
CNN	0.990	0.985
Majority	1.000	0.999

In our system, when a message is received from a user (specifically, in our case, from a Telegram user), we first check if the user is already known in our database (DB) containing all application users.

If the user is unknown, we add their information to our DB. Subsequently, we analyze the message using all available algorithms, gathering the results from each algorithm [4].

These individual results are then passed to the Majority Algorithm, which calculates the spam percentage of the message. The output of the Majority Algorithm is then transmitted to the Spam Analyzer, which determines whether the user who sent the message should be classified as a spammer. This determination is based on the calculated spam percentage of the message, along with the two most recent predictions. To identify a user as a spammer, we analyze their three most recent messages and compare the average spam percentage against a specified threshold. If the average spam percentage exceeds the threshold, we recognize the user as a spammer and record their ID in the DB of spammers. The proposed complex majority algorithm, illustrated in Fig. 6, utilizes the solutions obtained from the Bayesian spam filtering method, Perceptron, Support Vector Machine, and Convolutional Neural Network algorithms as inputs for the majority scheme.

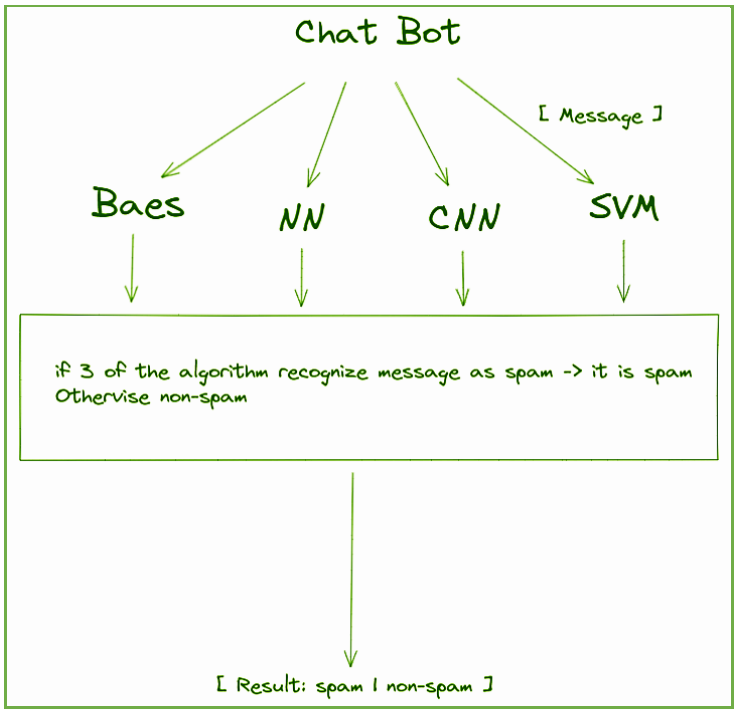

Fig. 6. The majority algorithm process

To align the algorithmic block outputs (ranging from 0 to 1) with the inputs of the majority scheme (binary values of 0 or 1), a binarization process is applied using a threshold of 0.95. The implementation of the spam analyzing and spammer analyzing is shown in Fig. 7.

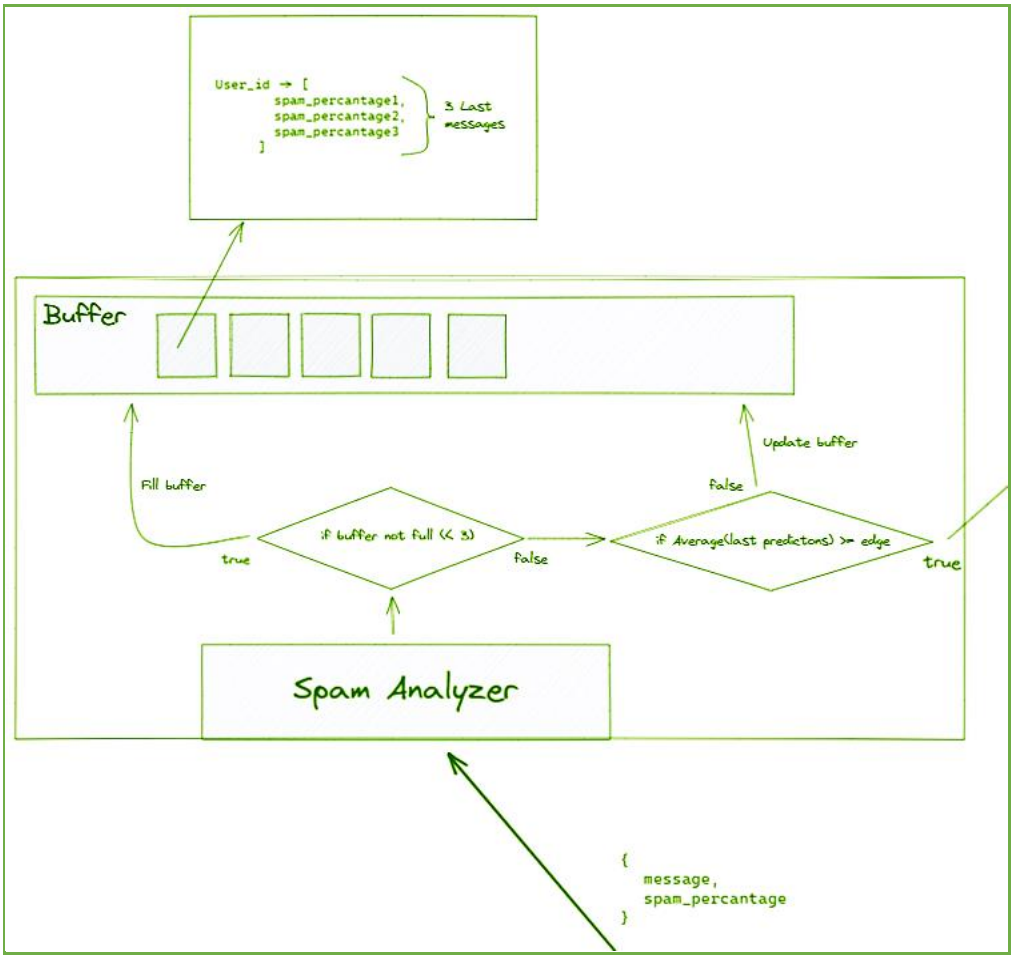

Fig. 7. The implementation of the spam analyzing and spammer analyzing

If a user is in the spammers DB his messages are being deleted without even analyzing them. The user receives the message that he was blocked. Only the manager of the application is able to remove users from the spammers.

The process of putting spammers to the DB and communication of the spam analyzer with the DB is shown in Fig. 8.

The general scheme of execution of the developed software application is given in Fig. 9 [20, 21].

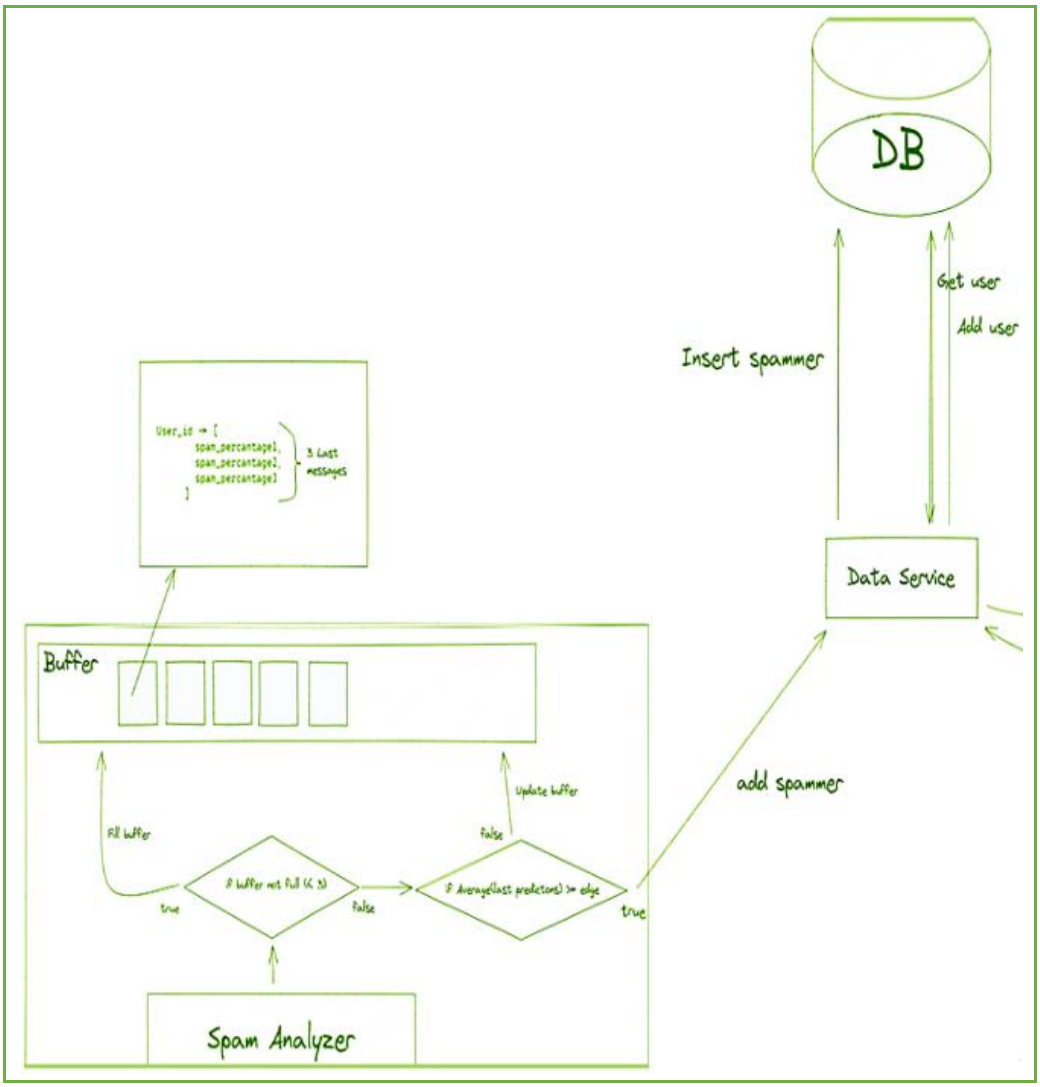

Fig. 8. The process of putting spammers to the DB and communication of the spam analyzer with the DB

Algorithm of analyzing spam messages contains the following steps:

1) the user enters into the software application the initial text that should be analyzed;

2) software application parses the initial text into array of words, then each word is converted to the infinitive, then the resulting set of words is vectorized and transmitted to the input to the all of the used algorithms;

3) the algorithms analyze the received data and returns the result as the probability of belonging the received data to the class (each algorithm has two classes: spam and non-spam);

4) the received data passed through the Majority Algorithm to calculate the spam percentage;

5) the app decides if the user should be marked as spammer based on the last 3 spam prediction of his messages;

6) if the user was identified as a spammer he is blocked.

Conclusions

This research project focused on addressing the scientific and applied problem of identifying spam within the textual context of social networking messengers, specifically utilizing the Kaggle SMS Spam Collection Dataset and employing chatbots in the popular messenger Telegram as an example.

The study encompassed the following key aspects:

1. Recognizing the relevance of spam detection and examining potential issues arising from spam interference.

2. Analyzing fundamental spam recognition methods, namely the naive Bayesian classifier, support vector method, multilayer perceptron neural network, and convolutional neural network.

3. Investigating fundamental approaches to detecting spammers.

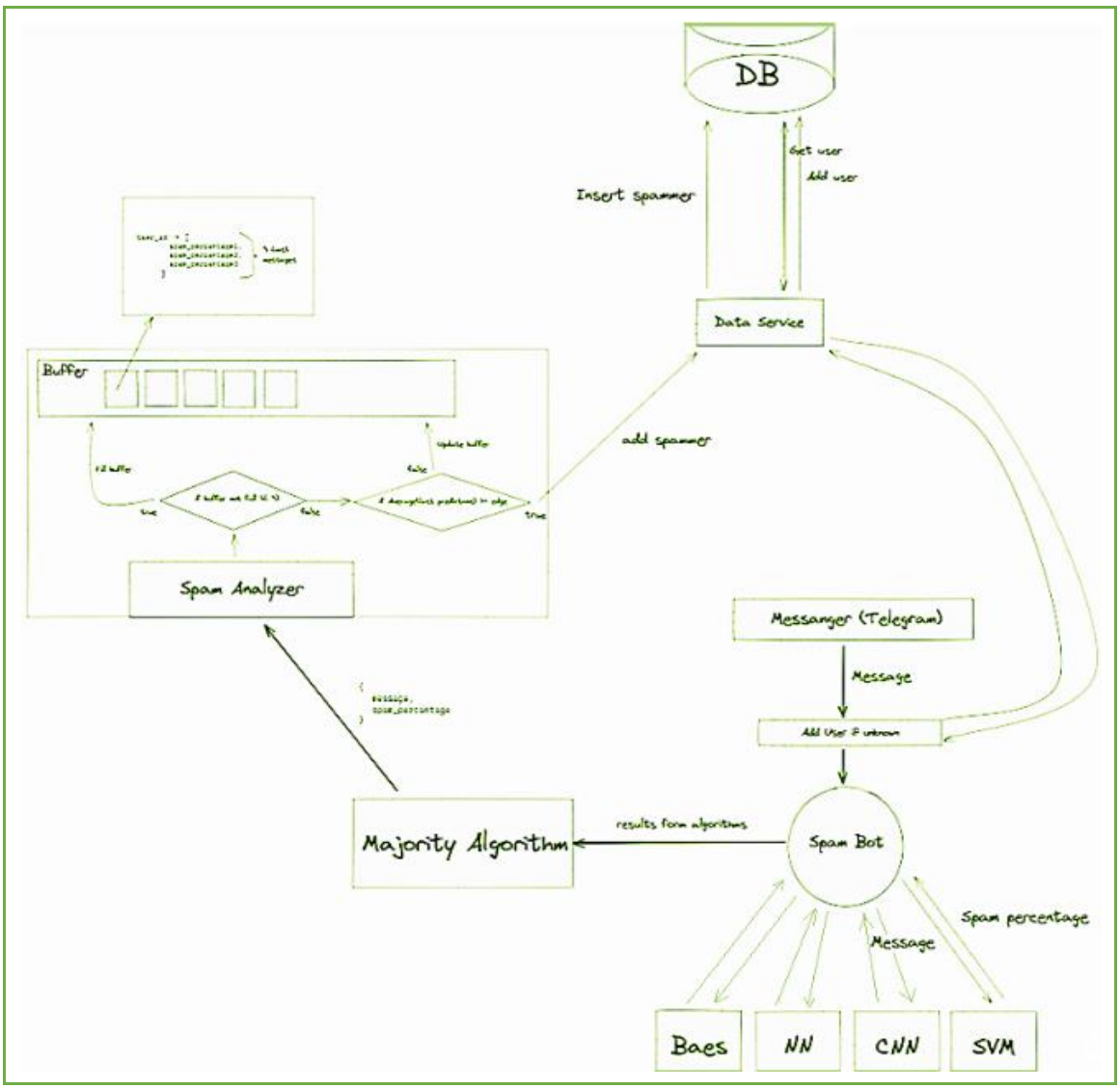

Fig. 9. The implementation of the spam analyzing and spammer analyzing

4. Developing a program designed to filter spam and detect spammers within the Telegram messenger. The program incorporated four implemented algorithms for spam recognition, along with a proposed complex majority algorithm. Furthermore, all text traffic was thoroughly inspected to identify potential spammers.

By addressing these elements, the research project successfully tackled the scientific and practical challenge of spam detection within social networking messengers, using the Kaggle SMS Spam Collection Dataset and implementing chatbots within Telegram as a case study.

References

1. Yasin, S. M. and Azmi, I. H. (2023), "Email spam filtering technique: challenges and solutions", *Journal of Theoretical and Applied Information Technology*, 2023, vol. 101, iss. 13, pp. 5130–5138.
2. Liu, B., Blasch, E., Chen, Y., Shen, D. and Chen, G. (2013), "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier", *Proceedings of the IEEE International Conference on Big Data*, 2013, USA, pp. 99-104. doi: https://doi.org/10.1109/BigData.2013.6691740.
3. Chaudhry, S., Dhawan, S., and Tanwar, R. (2020), "Spam Detection in Social Network Using Machine Learning Approach", *Data Science and Analytics. REDSET 2019. Communications in Computer and Information Science*, 2020, vol. 1230, pp. 236-245. doi: https://doi.org/10.1007/978-981-15-5830-6_20.
4. Liubchenko, N., Podorozhniak, A., Oliinyk, V. (2021), "Research of antispam bot algorithms for social networks", *CEUR Workshop Proceedings*, vol. 2870, 2021, pp. 822– 831, available at: http://ceur-ws.org/Vol-2870/paper61.pdf.
5. Sarkar, S. D., Goswami, S., Agarwal, A. and Aktar, J. (2014), "A Novel Feature Selection Technique for Text Classification Using Naive Bayes," *Int. Scholarly Research Notices*, 2014, article no. 717092. doi: https://doi.org/10.1155/2014/717092.
6. McCallum, A. and Nigam, K. (1998), "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI 1998: Learning for Text Categorization*, pp. 41-48, available at: http://courses.washington.edu/ling572/papers/mccallum1998_AAAI.pdf.
7. Zhang, W., Gao, F. (2013), "Performance analysis and improvement of naïve Bayes in text classification application," *Proceedings of the IEEE Conference Anthology*, China, pp. 1-4. doi: https://doi.org/10.1109/ANTHOLOGY.2013.6784818.

8. Nguyen, L. (2017), "Tutorial on Support Vector Machine," *Applied and Computational Mathematics*, vol. 6, pp. 1-15, available at: https://article.sciencepublishinggroup.com/pdf/10.11648.j.acm.s.2017060401.11.pdf.
9. Sastry, P. S. (2003), *An Introduction to Support Vector Machines*, 49 p., available at: http://www2.cs.uh.edu/~ceick/DM/Sastry_svm_notes.pdf.
10. Sharma, S. (2017), *What is the Perceptron*, available at: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53.
11. Chollet, F. (2021), *Deep learning with python*, Second Ed., Manning Publications, 504 p.
12. Deep, A.I. (2023), *Perceptron*, available at: https://deepai.org/machine-learning-glossary-and-terms/perceptron.
13. LeCun, Y., Bottou, L., Bengio, Y. and Haffner P. (1998), "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278 – 2324, doi: https://doi.org/10.1109/5.726791.
14. Yaloveha, V., Hlavcheva, D. and Podorozhniak, A. (2019), "Usage of convolutional neural network for multispectral image processing applied to the problem of detecting fire hazardous forest areas", *Advanced Information Systems*, 2019, vol. 3, no. 1, pp. 116-120, doi: https://doi.org/10.20998/2522-9052.2019.1.19.
15. Liubchenko, N., Podorozhniak, A. and Oliinyk, V. (2022), "Research Application of the Spam Filtering and Spammer Detection Algorithms on Social Media," *CEUR Workshop Proceedings*, vol. 3171, 2022, pp. 116-126, available at: https://ceur-ws.org/Vol-3171/paper13.pdf.
16. Masood, F., Ammad, G., Almogren, A., Abbas, A., and Zuair, M. (2019), "Spammer Detection and Fake User Identification on Social Networks," *IEEE Access*, 2019, vol. 7, pp. 68140-68152, doi: https://doi.org/10.1109/ACCESS.2019.2918196.
17. SMS Spam Collection Dataset [Data set], available at: https://www.kaggle.com/uciml/sms-spam-collection-dataset.
18. Python for Beginners. Python Software Foundation, available at: https://www.python.org/about/gettingstarted/.
19. Applications for Python. Python Software Foundation, available at: https://www.python.org/about/apps/.
20. Oliinyk, V., Podorozhniak, A. and Liubchenko, N. (2020), "Method of comprehensive spam recognition in social networks," *Proceedings of the 8th international scientific and technical conference Problems of informatization*, Ukraine, Vol. 2, p. 39, available at: http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/52856/1/Oliinyk_Method_comprehensive_2020.pdf.
21. Oliinyk, V., Podorozhniak, A. and Liubchenko, N. (2021), "Method of comprehensive spam recognition in social networks", *Proc. of the 8th int. scientific and technical conference Problems of informatization*, Ukraine, Vol. 1, p. 46, available at: http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/54913/1/Conference_NTU_KhPI_2021_Problemy_informatyzatsii_Ch_1.pdf.

Надійшла (received) 30.06.2023
Прийнята до друку (accepted for publication) 05.08.2023

Подорожняк Андрій Олексійович – кандидат технічних наук, доцент, доцент кафедри комп'ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Andrii Podorozhniak – Candidate of Technical Sciences, Associate Professor, Associate Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: andrii.podorozhniak@khpi.edu.ua; ORCID ID https://orcid.org/0000-0002-6688-8407.

Любченко Наталія Юріївна – кандидат технічних наук, доцент, доцент кафедри системного аналізу та інформаційно-аналітичних технологій, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Nataliia Liubchenko – Candidate of Technical Sciences, Associate Professor, Associate Professor of Systems Analysis and Information-Analytical Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: nataliia.liubchenko@khpi.edu.ua; ORCID ID https://orcid.org/0000-0002-4575-4741.

Олійник Василь Максимович – студент кафедри комп'ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Vasyl Oliinyk – master student of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: vasyl.oliinyk@cit.khpi.edu.ua; ORCID ID https://orcid.org/0000-0002-7582-3568.

Рог Вікторія Євгенівна – старший викладач кафедри протидії кіберзлочинності, Харківський національний університет внутрішніх справ, Харків, Україна;
Viktoriia Roh – Senior Lecturer of Combating Cybercrime Department, Kharkiv National University of Internal Affairs, Ukraine;
e-mail: vitochkarog@gmail.com; ORCID ID http://orcid.org/0000-0002-7443-5125.

Дослідження застосування алгоритмів фільтрації спаму та виявлення спамерів у соціальних мережах та месенджерах

А. О. Подорожняк, Н. Ю. Любченко, В. М. Олійник, В. Є. Рог

Анотація. Сьогодні існує багато різних соціальних мереж і месенджерів, які в часи пандемії коронавірусу та російської війни в Україні займають справді велику частину всього нашого життя, особливо в роботі. Крім того, проблема зі спамом і спамерами є як ніколи актуальною, кількість спаму в робочому текстовому потоці постійно збільшується. Під спамом ми розуміємо текстовий вміст, який не є необхідним у конкретному текстовому потоці, у випадку спамера мається на увазі особа, яка надсилає спам-повідомлення у своїх цілях. Стаття призначена для вирішення науково-прикладної проблеми виявлення спамерів та ідентифікації спам-повідомлень у текстовому контексті будь-якої соціальної мережі чи месенджера з використанням різних алгоритмів виявлення спаму та підходів виявлення спамерів. Ми реалізували 4 алгоритми: алгоритм, що використовує наївний байєсівський класифікатор, опорно-векторну машину, багатошарову нейронну мережу перцептрона та згорткову нейронну мережу. Дослідження було проведено з метою впровадження алгоритму виявлення спаму, який легко інтегрувати в месенджер (у нашому випадку ми використали Telegram як приклад). Створений алгоритм розпізнає спам на основі контексту конкретного текстового потоку, видаляє спам-повідомлення та блокує спамера, доки один із менеджерів програми не розблокує користувача-спамера.

Ключові слова: спам; соціальна мережа; наївний байєсівський класифікатор; опорно-векторна машина; багатошарова персептронна нейронна мережа; згорткова нейронна мережа; виявлення спамерів,