Research of Antispam Bot Algorithms for Social Networks

Nataliia Liubchenko, Andrii Podorozhniak and Vasyl Oliinyk

National Technical University “Kharkiv Polytechnic Institute”, Kyrpychova str. 2, Kharkiv, 61002, Ukraine

Abstract

There are many social media and messengers in use today, because of the situation with the corona virus pandemic the social media have become an integral part of our daily lives, including work activities. However, there is a lot of unnecessary information that comes to users in large quantities, so the problem of dealing with spam messages on social networks and messengers is now very relevant. By spam we mean any messages that a particular user (person, company, etc.) considers unnecessary in a particular text stream. The project is dedicated to solving the scientific problem of detecting spam messages in the text context of any social network or messenger using anti-spam bot that is based on various spam detection algorithms. Four algorithms were implemented and investigated: an algorithm using naive Bayesian classifier, support vector method, multilayer perceptron neural network and convolutional neural network. The main idea is to develop a complex spam detection algorithm for anti-spam bot, which is fast and easy to implement in a messenger (social network). We propose to use the application of the obtained solutions for IT companies. The developed complex algorithm can be used not only to remove spam, but also, for example, to monitor chats for messages that are important to a particular user.

Keywords 1

Spam, social network, antispam bot, support-vector machine, convolutional neural network

1. Introduction

In 2019, the percentage of spam in global mail traffic was 56.51%, which is 4.03% more than in 2018 (Figure 1) [1].

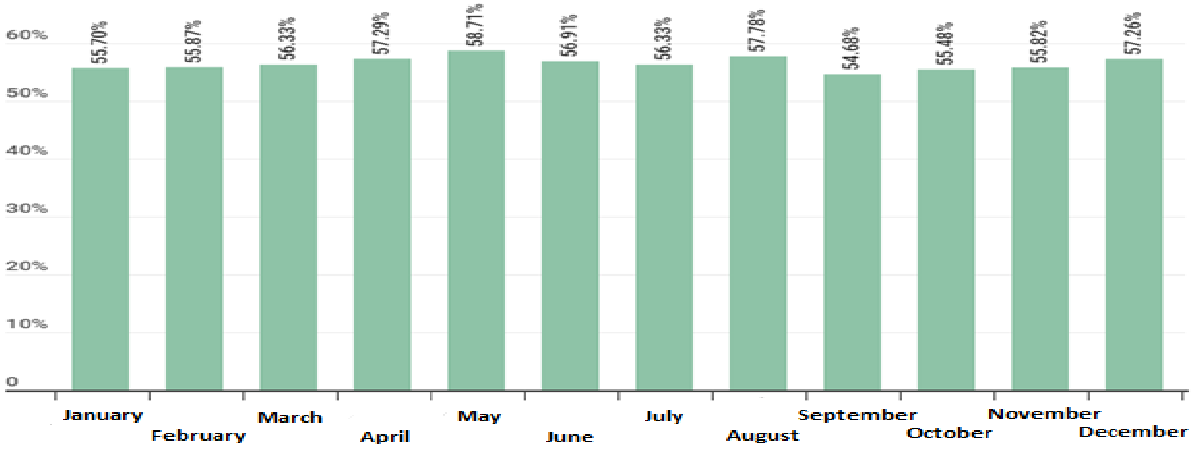


Figure 1: Percentage of spam in email traffic in 2019

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: nlubchenko63@gmail.com (N. Liubchenko); andriipodorozhniak@gmail.com (A. Podorozhniak); oleynikwasya@gmail.com (V. Oliinyk)
ORCID: 0000-0002-4575-4741 (N. Liubchenko); 0000-0002-6688-8407 (A. Podorozhniak); 0000-0002-7582-3568 (V. Oliinyk)

Unlike inboxes, most social media chat rooms do not have built-in anti-spam algorithms. Although there is not as much spam in chat rooms as in email services, the cost of an extra message can cost a particular firm a lot of money. For example, when a spam message contains a link to a resource that infects a user's computer and penetrates a company's internal network. The effects of such interference can be very costly for a firm. Therefore, there is the issue of monitoring the incoming text stream in social networks and messengers.

By being able to filter spam messages in messengers and social networks, firms can save their employees' time and prevent losses information.

To solve the problem we were used algorithms using support vector method, convolutional neural network and naive Bayesian classifier. An approach with integrated application of the investigated algorithms can begin to solve the problem of spam in social networks and messengers.

2. Characterization of spam and how to deal with it

Spam is a mass mailing of correspondence of an advertising or other types of spam, to the people who have not expressed a desire to receive it [2]. The various types of spam generally include: advertisements; Nigerian emails; phishing; and other types of spam. Other types of spam include: mass mailings of letters with religious content; mass mailings to put the mail system out of operation (to bring the system into service failure); mass mailings on behalf of another person in order to cause a negative attitude towards him or her; mass mailings of letters containing computer viruses (for their initial distribution).

The basic ways of spam distribution today include [3, 4]: e-mail; Usernet; messengers; spoofing of Internet traffic; SMS messages; phone call, etc.

Spam messages cost the perpetrator virtually nothing, but the recipient of the spam usually has to pay the ISP for the time used to receive the spam. Also, the massive proliferation of spam complicates information systems and resources, with a very large amount of unnecessary loading.

Due to the mass mailings, users are forced to spend extra time filtering messages. To avoid this, users use anti-spam filters to save time. But spam filters can also accidentally erase an important message by recognizing it as spam.

The best way to deal with spam is to prevent spammers from getting hold of your email address.

Auto-Spam detection software is called anti-spam filters. They can be used by end-users or on servers. This software has two main approaches [5].

1. The content of the message is analyzed, based on that it is concluded whether it is spam or not. If a message is classified as spam, it can be flagged, moved to another folder or even deleted. Such software can run both on the server and on the client computer. With this approach you don't see the spam filtered, but you continue to pay the full cost for receiving it, because the anti-spam software receives each spam message anyway (wasting your money) and only then decides whether to show it or not.
2. It classifies the sender as a spammer without looking at the text of the message. This software can only work on the server which directly receives the messages. With this approach it's possible to reduce the cost - money is only spent on communicating with spam mailing programs (i.e. refusing to accept the messages) and on contacting other servers (if any) for verification. The gain, however, is not as great as you might expect. If the recipient refuses to accept the message, the spammer program tries to bypass the protection and send it another way. Each such attempt has to be repelled separately, which adds to the load on the server.

Existing methods of combating spam are based on known technologies of the classical Bayesian approach [6], the support vector machines [7], the use of deep learning [8], etc., and on the combination of known methods [4, 9] or the development of new approaches [10].

3. Results and discussions

In this project we have discussed: the statistical Bayesian spam filtering method, application of support vector machines, multilayer perceptron neural network and convolutional neural network to create a complex anti-spam bot algorithm in social networks [11, 12].

To implement the spam filtering algorithms we used Python 3.6 programming language, PyCharm programming environment and Keras, NumPy, Sklearn and Pandas libraries [13, 14]. The simulation was performed on a LifeBook E744 laptop with 8Gb RAM, an Intel Core i7 CPU (up to 3.2 GHz) and an Intel HD Graphics 4600 video processor.

The estimation of the probability of correct spam recognition was calculated as the ratio of correctly recognized messages to the total number of messages (separately for training and test samples). The estimate of the probability of erroneous recognition of spam was calculated as the ratio of erroneously recognized messages to the total number of messages. Also, in addition to the estimation of the probability of correct spam recognition for selected and proposed algorithms, we have used an F1 assessment [15, 16].

3.1. Data preparation

The spam message dataset from the kaggle SMS Spam Collection Dataset [17] was chosen as the training dataset.

The Spam messages dataset contains 5162 messages (of which 13% are spam), arranged in two columns, the first of which is the message class (ham – not spam messages, spam - spam messages).

First, convert the first class column to an array of integers, where 1 indicates that the given message is spam, 0 indicates a regular message.

In order to turn the whole message text into numbers (vectorizable), we need to build a word dictionary, an array of words [18, 19]. Each word in the dataset is converted to lower case. Since our set could get punctuation marks, the next step is to clear the dictionary of unnecessary words.

After that each word is brought to infinitives (fish, fishing -> fish).

Analyzing the resulting set of data we can see some words are repeated very often, and others, on the contrary, very rarely. Delete from the resulting set of words the words that are repeated more than 1000 and less than 5 times.

On the basis of the obtained set of words, let's create a dictionary, deleting all repetitions (i.e. each word in the dictionary is unique). For this purpose we used the vectorizer from sklearn library.

The created dictionary includes 1191 words [12]. Vectorize dataset sentences with the dictionary.

Let's look at this process in more detail. For example, we have a vocabulary with the words:

I, buy, cat, dog, yesterday, myself.

Then the sentence "Yesterday I bought myself a dog" will be vectorized as:

[1, 1, 0, 1, 1, 1],

and the sentence "I bought a cat":

[1, 1, 1, 0, 0, 0].

After conversion, the data can be passed to the input classifiers. Also some part of the dataset (about 600 messages), which is not part of the training data, was allocated for further testing of the algorithms.

By testing each of the algorithms, we will count the number of errors and the recognition error percent.

The spam message analysis algorithms contain the following steps.

1. The user enters into the software application the source text to be analyzed
2. The software application splits the original text into words, then each word is brought to its original state (infinitive), then the resulting set of words is vectorized and sent to the input of the algorithm
3. The algorithm analyses the received data and returns the result as a probability of class membership of the received data (our algorithm has two classes: spam and non-spam)
4. The obtained data from the algorithm is analyzed and transformed to a user-understandable representation
5. The software application displays the result of the analysis of the received message, as well as the probability with which the message was recognized and the name
6. If the user wishes, the result of the analysis can be written to the file.

3.2. Testing algorithms

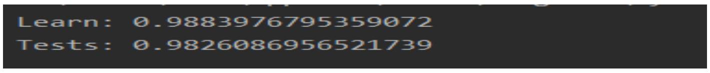
A naive Bayesian classifier is a probabilistic classifier that uses Bayes' theorem to determine the probability of an observation (sample item) belonging to one of the classes, assuming (naive) independence of the variables [20]. The use cases of this method can be: recognition of spam, analysis of emotional coloring of texts, identification of racism in a text selector, any information processing system and so on.

So, if based on the values of the variables it can be uniquely determined to which class an observation belongs, the Bayesian classifier will report the probability of belonging to that class [21].

In cases, where an observation can belong to different classes with different probabilities, the result of the classifier will be a vector whose components are the probabilities of belonging to one class or another [22].

The advantage of this approach is that the sample size requirements are reduced from exponential to linear. The disadvantage is that the model is accurate only when the assumption of independence is satisfied. Otherwise, strictly speaking, the calculated probabilities are no longer accurate (and even more, their sum may not equal one, making it necessary to normalize the result).

The results of the Bayesian classifier of correct spam detection for the training and test samples are shown in Figure 2.



```
Learn: 0.9883976795359072
Tests: 0.9826086956521739
```

Figure 2: The Results of the Bayesian classifier recognition.

In machine learning, Support Vector Machines (SVM) is a method of data analysis for classification and regression analysis using supervised learning models with associated learning algorithms, called support vector machines [23, 24]. For a given set of training samples, each marked as appropriate to one or the other of two categories, the SVM training algorithm builds a model that assigns new samples to one or the other category, making it a probabilistic binary linear classifier. The SVM model is a representation of samples as points in space, reflected in such a way that samples from individual categories are separated by a net gap, which is the widest. New samples are then mapped into the same space and predictions are made about their category membership based on which side of the gap they fall on Figure 3.

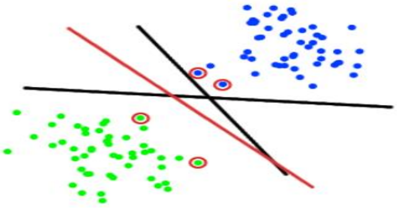


Figure 3: The result of the method when data is not linearly separate

But there are situations when the data cannot be separated linearly. For this reason, it was proposed to map the primary finite-dimensional space to a space of much higher dimension, presumably making the separation simpler in that space Figure 4.

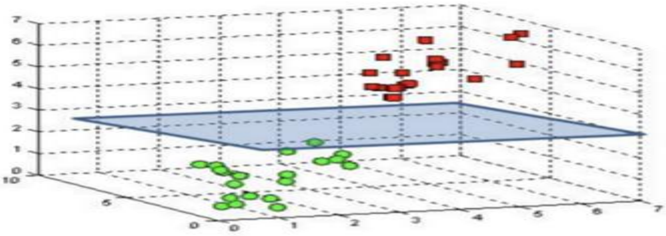
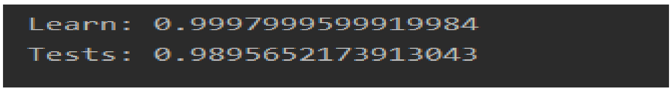


Figure 4: The result of the method when data is linearly inseparable

The results of the SVM in the form of estimating the probability of correct spam recognition for training and test samples are shown in Figure 5.



```
Learn: 0.9997999599919984
Tests: 0.9895652173913043
```

Figure 5: The results of the work for the method of reference vectors

Perceptron is a mathematical or computer model of information perception by the brain (cybernetic model of the brain), proposed by Frank Rosenblatt in 1957 and implemented in the form of an electronic machine "Mark-1". The main mathematical problem he is able to cope with is the linear separation of arbitrary nonlinear sets [18].

The perceptron consists of three types of elements, namely: the signals coming from the sensors are transmitted to the associative elements, and then to the responders. Thus, perceptrons allow you to create a set of "associations" between the input stimuli and the required response at the output. Biologically, this corresponds to the transformation, for example, of visual information into a physiological response of motor neurons.

In our case, a perceptron multilayer neural network was used, which consists of 4 layers, namely: 1 input, 2 hidden and 1 output, as shown in Figure 6.

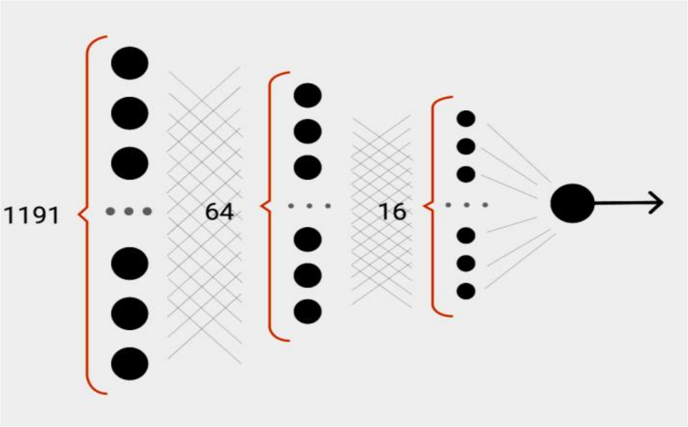


Figure 6: Schematic model of the applied perceptron neural network

Estimation of the probability of spam detection in the test sample depending on the number of epochs of learning the perceptron neural network is shown in Figure 7.

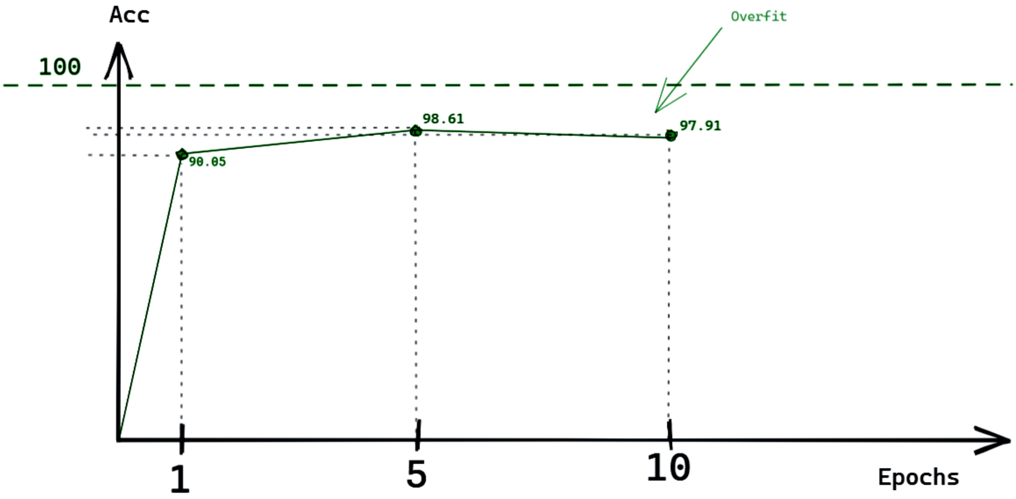
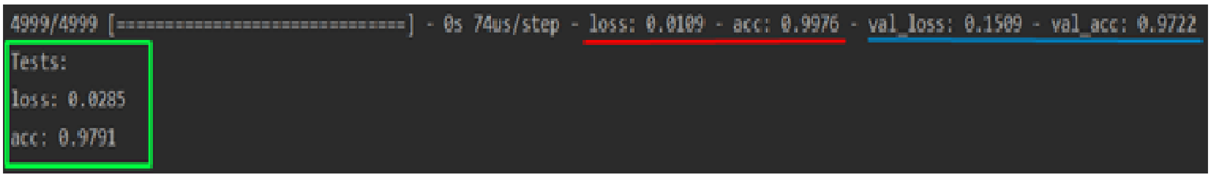


Figure 7: The results of the perceptron neural network depending on the number of learning epochs

The obtained results of the perceptron neural network (without the effect of retraining) in the form of estimating the probability of correct spam recognition for training and test samples are shown in Figure 8.



```
4999/4999 [==============================] - 0s 74us/step - loss: 0.0109 - acc: 0.9976 - val_loss: 0.1509 - val_acc: 0.9722
Tests:
loss: 0.0285
acc: 0.9791
```

Figure 8: The results of work for a multilayer perceptron neural network

Convolutional neural network, (CNN) – special architecture of artificial neural networks, proposed by Jan Lekun in 1988 [25] and aimed at effective pattern recognition, is part of Deep learning technologies. The structure of the network is unidirectional, without feedback, fundamentally multilayered.

The network architecture got its name due to the presence of the convolution operation, the essence of which is that each image fragment is multiplied by the convolution matrix (core) element by element, and the result is summed and recorded in a similar position of the original image [26].

The structure of the CNN we used is shown in Figure 9.

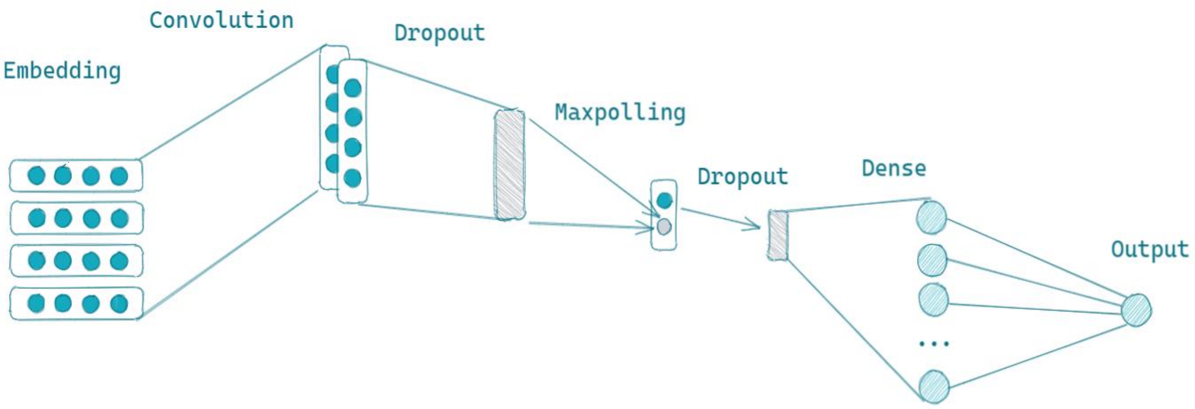


Figure 9: Schematic model of the applied CNN

An estimate of the probability of recognizing spam in a test sample depending on the number of epochs of CNN training is shown in Figure 10.

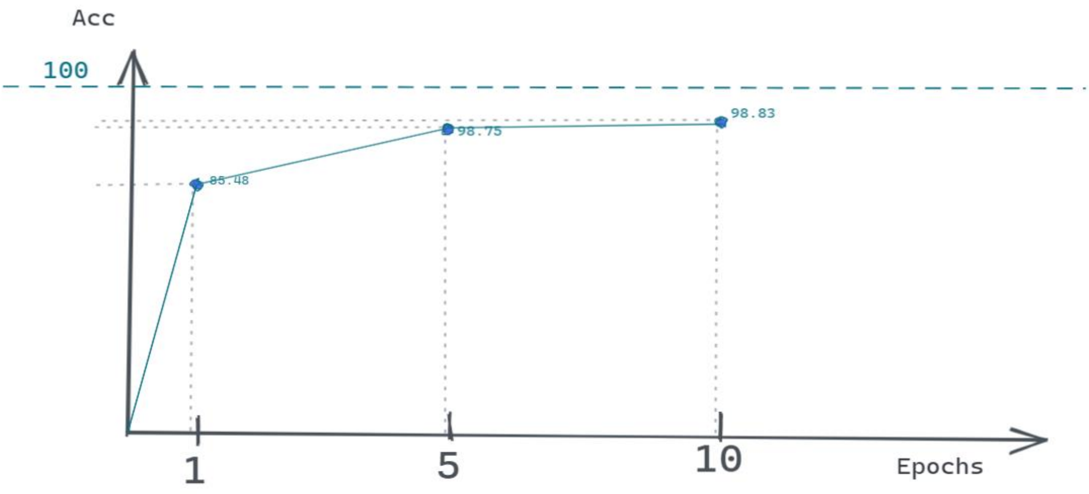
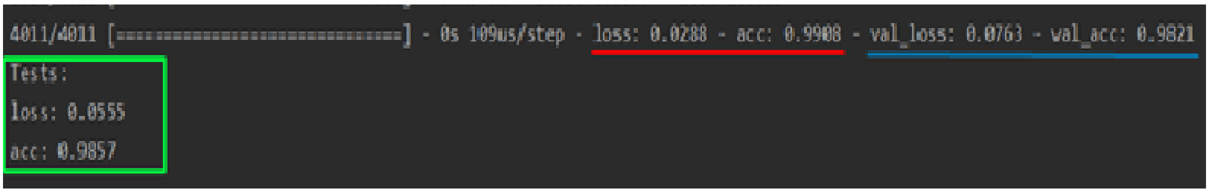


Figure 10: CNN results depending on the number of learning epochs

The results obtained by CNN (for 10 learning epochs) in the form of an estimate of the probability of correct spam recognition for training and test samples are shown in Figure 11.



```
4011/4011 [==============================] - 0s 109us/step - loss: 0.0288 - acc: 0.9908 - val_loss: 0.0763 - val_acc: 0.9821
Tests:
loss: 0.0555
acc: 0.9857
```

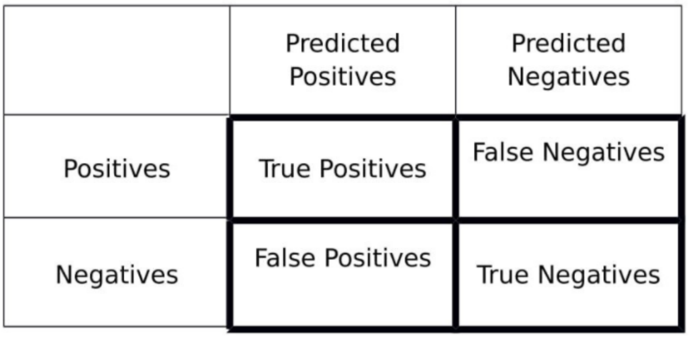
Figure 11: CNN results (for the 10th epoch of learning)

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

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.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{1}$$

where *TP* – (True Positive) correctly classified positive sample;
FN – (False Negative) the sample is positive but it is classified as negative;
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.



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

Figure 12: The explenation of accurancy evaluation

The results of the tests using accurancy metric are shown at the Table 1.

Table 1
The results of the testing algorithms on training and test samples

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

3.3. Antispam bot algorithm

After analyzing the results of testing the statistical Bayesian method of spam filtering, the method of reference vectors, multilayer perceptron neural network and convolutional neural network, the three best accuracy algorithms were selected: Bayesian method of spam filtering, the method of reference vectors and convolutional.

For the anti-spam bot on the social network, we propose a complex algorithm, which includes the parallel operation of the selected algorithms with the decision on the presence of spam by a majority scheme of two out of three, as shown in Figure 13.

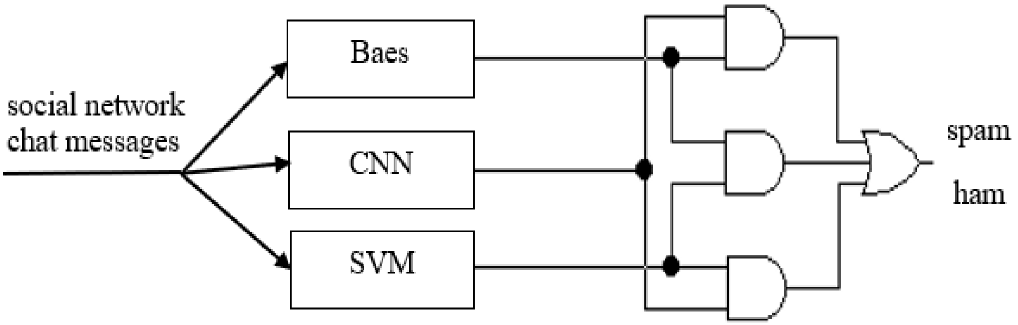
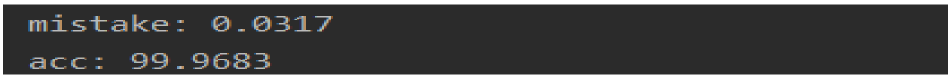


Figure 13: Scheme of the complex majority algorithm of antispam bot operation

The proposed complex algorithm shown in Figure 13 uses as inputs for majority scheme the solutions of the Bayesian spam filtering method, support vector method and convolutional neural network algorithms. To match the outputs of the algorithmic blocks (0… 1) with the inputs of the majority scheme (0, 1), their binarization with a threshold of 0.95 is performed. The results of the complex algorithm of antispam bot in the form of an estimate of the probabilistic of correct spam recognition for the test samples are shown in Figure 14.



```
mistake: 0.0317
acc: 99.9683
```

Figure 14: The results of recognition of the complex algorithm of antispam bot

The probability of correct spam recognition for the proposed antispam bot algorithm is much better than the results obtained for each of the researched most popular algorithms separately. Perhaps such a large difference is due to relatively small sample data for training and testing

The antispam bot’s analysis time of each of the tested messages was less than 0.5 sec, which allows it to be used in real-time systems.

The main problem with classification accuracy in multiclass tasks is that it is sensitive to the imbalanced data. It is possible to get high results when the number of samples in one class more than in others, but it doesn’t mean that in a real world a neural network will classify all classes correctly with the same value of success.

F1 score is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.

The defenition of precision, recall and F1 score made in accordance with the formulas [15]:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{4}$$

The results of the tests using F1 score are shown at the Table 2.

Table 2

The results of the testing algorithms on training and test samples

Algorithm	F1	Recall	Precision	Accuracy
Bayes	0.927	0.990	0.864	0.982
SVM	0.951	0.971	0.932	0.989
NN	0.962	0.928	0.928	0.979
CNN	0.962	0.950	0.925	0.985
Majority	0.965	0.998	0.932	0.999

In the F1 score metric, the proposed majority algorithm also showed the best results.

4. Conclusions

As part of this research, the scientific and applied problem of research and development antispam bot algorithm for the textual context of social networking messengers was solved by the example of Kaggle SMS Spam Collection Dataset using chatbots in the popular messenger Telegram.

Considered the relevance of spam detection and possible problems due to spam intervention.

A comparative analysis of the three most popular methods for recognizing spam messages was performed and it was shown that the most effective method is the support vectors algorithm with accuracy only 0.989. In second place was the algorithm based on the convolutional neural network with accuracy 0.985. In third place was the algorithm based on the naive Bayes classifier with accuracy 0.982. The fourth place for our case was taken by the multilayer perceptron neural network with accuracy 0.979.

It was developed the program to filter spam in the messenger Telegram, that uses the majority combination three implemented better algorithms for spam recognition. The created antispam bot has accuracy 0.999, F1 score – 0.965 and can be used in real-time systems.

5. References

[1] M. Vergelis, T. Shcherbakova, T. Sidorina, T. Kulikova, Spam and phishing in 2019, April 8, 2020. URL: https://securelist.ru/spam-report-2019/95727.

[2] S. Chaudhry, S. Dhawan, R. Tanwar, Spam Detection in Social Network Using Machine Learning Approach, in: U. Batra, N. Roy, B. Panda (Eds.), Data Science and Analytics. REDSET 2019, Communications in Computer and Information Science, 2020, pp. 236-245. doi:10.1007/978-981-15-5830-6_20.

[3] R. Krithiga1, E. Ilavarasan, A Comprehensive Survey of Spam Profile Detection Methods in Online Social Networks, Journal of Physics: Conference Series, 1362, 012111, 2019. doi:10.1088/1742-6596/1362/1/012111.

[4] Ch. Zhao, Y. Xin, X. Li, Y. Yang, Yu. Chen, A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data, Applied Sciences, 10, 936, 2020. doi:10.3390/app10030936.

[5] Ways to combat spam, Ostriv znan, July 29, 2008. URL: http://korysne.ostriv.in.ua/publication/code-24F002FC35B8C/list-1420E79CF27.

[6] Y. Begriche, H. Labiod, A Posterior Distribution for Anti-Spam Bayesian Statistical Model, Conference on Network and Information Systems Security, La Rochelle, France, 2011, pp. 1-6. doi: 10.1109/SAR-SSI.2011.5931393.

[7] O. Amayri, N. Bouguila, A study of spam filtering using support vector machines, Artificial Intelligence Review, 34, 73–108, 2010. doi:10.1007/s10462-010-9166-x.

[8] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, O. E. Ajibuwa, Machine learning for email spam filtering: review, approaches and open research problems, Heliyon, 5, e01802, 2019. doi: 10.1016/j.heliyon.2019.e01802.

[9] P. Navaney, G. Dubey, A. Rana, SMS Spam Filtering Using Supervised Machine Learning Algorithms, in: Proceedings of the IEEE 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), India, NSPEC Accession Number: 18044645, 2018. doi:10.1109/CONFLUENCE.2018.8442564.

[10] Yu. Parzhin, V. Kosenko, A. Podorozhniak, O. Malyeyeva, V. Timofeyev, Detector neural network vs connectionist ANNs, Neurocomputing, Vol. 414, 2020, pp. 191–203. doi: https://doi.org/10.1016/j.neucom.2020.07.025.

[11] V. Oliinyk, A. Podorozhniak, N. Liubchenko, Method of comprehensive spam recognition in social networks, in: Proceedings of the 8th international scientific and technical conference Problems of informatization, Ukraine, Vol. 2, p. 39, 2020. URL: http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/50565/1/Conference_NTU_KhPI_2020_Problemy_informatyzatsii_Ch_2.pdf.

[12] V. Oliinyk, N. Liubchenko, A. Podorozhniak, Research of the method of complex spam recognition in social networks, in: Proceedings of the XIV International scientific-practical conference of undergraduates and graduate students "Theoretical and practical research of young scientists", Kharkiv, Ukraine, p. 8, 2020. URL: http://web.kpi.kharkov.ua/masters/wp-content/uploads/sites/135/2020/12/ TPRYS-2020.pdf.

[13] H. Lane, H. Hapke, C. Howard, Natural Language Processing in Action: Understanding, analyzing, and generating text with Python, Manning Publication, New York, NY, 2019.

[14] Applications for Python, Python Software Foundation, 2019. URL: https://www.python.org/about/apps/.

[15] A. S. Desuky, S. An Improved Hybrid Approach for Handling Class Imbalance Problem, Arabian Journal for Science and Engineering, pp. 3853–3864, 2021. https://doi.org/10.1007/s13369-021-05347-7

[16] H. Dalianis, Evaluation Metrics and Evaluation, In: Clinical Text Mining, Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-78503-5_6.

[17] SMS Spam Collection Dataset [Data set], 2017. URL: https://www.kaggle.com/uciml/sms-spam-collection-dataset.

[18] F. Chollet. Deep learning with python, Manning Publications, New York, 2018.

[19] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, London, 2017.

[20] W. Zhang, F. Gao, Performance analysis and improvement of naïve Bayes in text classification application, in: Proceedings of the IEEE Conference Anthology, China, pp. 1-4, 2013. URL: https://doi.org/10.1109/ANTHOLOGY.2013.6784818.

[21] B. Liu, E. Blasch, Y. Chen, D. Shen, G. Chen, Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier, in: Proceedings of the IEEE International Conference on Big Data, USA, pp. 99-104, 2013. doi:10.1109/BigData.2013.6691740.

[22] A. McCallum, K. Nigam, A Comparison of Event Models for Naive Bayes Text Classification, AAAI 1998: Learning for Text Categorization, pp. 41-48, 1998. URL: http://courses.washington.edu/ling572/papers/mccallum1998_AAAI.pdf.

[23] L. Nguyen, Tutorial on Support Vector Machine, Applied and Computational Mathematics, 6 (4), 1–15, 2017. doi:10.11648/j.acm.s.2017060401.11.

[24] N. Sharma, Understanding the Mathematics behind Support Vector Machines, Heartbeat, 2020. URL: https://heartbeat.fritz.ai/understanding-the-mathematics-behind-support-vector-machines-5e20243d64d5.

[25] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 86 (11), pp. 2278–2324, 1998.

[26] V. Yaloveha, D. Hlavcheva, A. Podorozhniak, Usage of convolutional neural network for multispectral image processing applied to the problem of detecting fire hazardous forest areas, Advanced Information Systems, 3, 1, pp. 116-120, 2019. doi:10.20998/2522-9052.2019.1.19.