DOI 10.20535/2411-1031.2023.11.2.293797
UDC 004.032.26

VOLODYMYR ONISHCHENKO,
ANATOLII MINOCHKIN

ANALYSIS OF METHODS OF CLASSIFICATION OF ELECTRONIC MESSAGES BASED ON NEURAL NETWORK MODELS

In the article, the creation of a mechanism for detecting and classifying messages is considered, with an assessment of how effectively different neural networks work and can recognize and classify different types of electronic messages, including phishing attacks, spam, and legitimate messages. A preliminary analysis of incoming messages has been performed, encompassing their headers, text, and other relevant attributes. For instance, in the case of emails, these attributes could be the 'subject' and 'sender' of the message. Methods for data preparation and processing have been reviewed, including text vectorization, noise removal, and normalization, to be utilized in training neural networks. Message tokenization has been performed by transforming them into a numerical format while considering the selection of features. For text messages, it is crucial to execute both tokenization and text vectorization. The model training was performed on the test data with prior splitting into two parts: 80% for training and 20% for testing. The training set is utilized for training the model, while the test set is used to evaluate its effectiveness. The peculiarity of the class structure of the data, namely the uniformity of the distribution of classes, is considered. In this case, spam occurs less frequently than legitimate messages, so class balancing techniques such as random deletion of redundant examples, upsampling, and subsampling were applied to ensure adequate model training. Optimization of network parameters was performed, by researching the optimal parameters of neural networks, such as the number and size of layers, activation functions, and optimization of hyperparameters to achieve the best performance. Hyperparameter optimization includes determining optimal settings for neural networks, such as layer size, activation functions, learning rate, and other parameters. The effectiveness was assessed by comparing the results and performance of various classification methods based on neural networks using metrics such as precision and F1-score. It was determined how well the methods can avoid misclassifications where legitimate messages are mistakenly identified as spam, and vice versa. A comparison of the methods' effectiveness in processing a large volume of messages in real time was conducted. An analysis of different architectures of neural network models was performed. Based on the analysis, it was revealed how effectively different neural network models can recognize and classify messages as spam.

Keywords: message classification, neural networks, natural language processing, spam filtering, text vectorization, email classification, text analysis, model quality evaluation.

Problem Statement: In the modern information environment, the processing and classification of electronic messages have become crucial due to the increasing volume of electronic communication. Despite significant progress in using neural networks to classify electronic messages, some key issues need attention and resolution. One of these issues is the insufficient classification effectiveness under conditions of architectural uncertainty, instability in working with insufficient data volumes, and difficulties in constructing models that are interpretable and protected from attacks. There are challenges and issues in the application of neural networks for the classification of electronic messages. Among them are the instability of results due to insufficient data volume, the complexity of selecting the optimal network architecture and parameters, as well as the difficulty in ensuring the interpretability of the accepted models for message analysis. Additionally, data balancing and model protection from attacks are important aspects to address. Further research and

improvement of classification methods based on neural networks are needed to ensure high accuracy and efficiency in real-world conditions. For the creation of an electronic message classification system, it is necessary to follow the sequence of actions, namely:

– Primary spam filtering, which includes detecting spam using special filters and analysis of message headers, content, and other characteristics.

– Categorization by message type, recognition of messages from personal contacts, and identification of messages containing advertising or commercial offers.

– Notifications and update messages that contain information about changes and events.

– Prioritization of important messages and distribution of messages based on the user's priority.

– Text analysis to detect mood (positive, negative, neutral).

– Distribution of messages by specific topics or categories.

– Detection of viruses and malicious content by analyzing messages, attachments, and links to identify potentially dangerous elements.

– Adaptation of the model by training a system that can consider the user's choice and personalize the classification.

In general, the classification of electronic messages involves several main components. One of the main components is the presence and quality of data. Data quality encompasses the volume and diversity of messages in the training set, as well as the adequacy of representing different classes of messages. The chosen classification method takes into account the specificity of the data. This could be a neural network, decision tree, support vector machine (SVM), or other machine learning algorithms. Feature engineering is crucial; effective classification requires appropriately chosen features or characteristics of the text that are meaningful for classification. These may include words, syntactic features, text structure, etc.

A good classifier should include data cleaning, tokenization, vectorization, and normalization. This stage helps transform texts or data into a format that is understandable by the classifier. The best classification algorithm for the task can be chosen: from simple models, such as a naive Bayes classifier, to more complex ones, such as convolutional or recurrent neural networks (CNN, RNN), or ensembles of models. Adjusting model parameters for optimal performance may involve optimizing the sizes of network layers, learning rates, and activation functions. It is crucial to evaluate the model on a test data set to check its accuracy and avoid overtraining; perform optimization of speed and resource efficiency, including selecting optimal algorithms for data processing speed and model operation; provide dropout data regularization or L1/L2 regularization. These techniques help prevent overtraining and improve the generalization capabilities of the model. Use the cross-validation approach by evaluating the model in conditions of limited data availability. Identifying class imbalance and applying class balancing methods (such as oversampling or undersampling) can improve classification effectiveness. Ensure the effectiveness of the model by monitoring and updating it over time. An effective classifier is a comprehensive tool that combines not only algorithms but also the entire process of data processing, tuning, and model improvement to achieve optimal accuracy and versatility.

An analysis of existing research and publications has shown that the classification of electronic messages can be solved using neural network models [1]. Currently, models based on the naive Bayesian classifier are actively used to solve the problem of classifying electronic messages. The naive Bayes classifier assumes the independence of features (words) in the text, which can be presented in the following form [2]:

$$P(d) = \sum_{i=1}^{k} P(d \mid c) \times P(c_i), \quad (1)$$

where d represents a vector of a textual document that needs to be classified based on its content. The vector contains information about the frequency of individual words in the text, TF-IDF values, and other text properties used for classification. TF-IDF (Term Frequency-Inverse Document Frequency) is a text vectorization method used to assess the importance of terms (words or phrases) in the context of electronic messages.

c represents a specific class ("spam" or "non-spam") or category to which the textual document d may belong. $P(d \mid c)$ is the probability of the text d belonging to class c. $P(c)$ is the probability that document d belongs to a specific class c without any additional information or context, based on general knowledge or the distribution of classes in the data sample. $P(d)$ is the overall probability of the text d.

k is the number of possible classes. According to formula (1), it is possible to determine that the given method does not consider values in different contexts.

Naive Bayes models do not consider context and may suffer from the problem of ambiguity, directly impacting the accuracy of estimation and the identification of unwanted messages. In large text corpora using a Naive Bayes model, the issue of sparsity [3] may arise when many words or features have low entry frequency. This can lead to inaccuracies in probability estimates for these words. In this analysis, it is evident that to improve spam filtering performance, it is necessary to increase the quality of classifying incoming messages by considering logical text sequences and balancing data by avoiding sparsity [4].

The purpose of the study is to analyze and compare the methods of classification of electronic messages based on neural network models, considering classification accuracy and model speed to enhance the quality of identifying harmful electronic messages in information exchange systems.

Formulation of the electronic message classification problem: The general task of text classification involves assigning one or several labels (classes) to textual documents based on their content or characteristics. Mathematically, this can be formulated as follows: Let there be a training set:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, \quad (2)$$

where x_n is a text document;

y_n is a label (class) to which it belongs;

n is the number of documents in the training set.

To solve the problem of text classification of electronic messages, it is necessary to build a model $f(x)$, which can predict a label y for a new text x. The task involves selecting a function f that can correctly assign class labels to new texts based on the training set. Mathematically, this can be expressed as building the model f using the training set to maximize the accuracy of predictions for new texts. Prediction accuracy in the context of text classification can be described as the ratio of correctly classified documents to the total number of documents considered by the model. Mathematically, this is expressed by the formula:

$$T = X_n / X_k, \quad (3)$$

where X_n is the number of correctly classified messages;

X_k is the total number of messages.

To maximize the accuracy of the neural network model predictions, it is necessary to choose optimal model parameters and loss functions to achieve the best classification results on the test dataset. In this case, classifying texts in electronic messages involves using models of the Naive Bayes classifier [7] and neural network models to assign labels (classes) to texts based on their content. The main stages of this process can be described as follows:

– Transforming the text xi into a sequence of tokens (words or phrases), which can be represented as:

$$x_i = (t_{i1}, t_{i2}, \ldots, t_{im}), \quad (4)$$

where m is the number of tokens in document;

t_{ij} – j-th token in document i.

– Choosing a machine learning model f for text classification. For example, f could be a Naive Bayes classifier, Support Vector Machine (SVM), neural network, etc.

– Training the model f on the training set D to predict labels y for new texts. This can be expressed as finding optimal model parameters for the best mapping between input texts and their labels.

– Using the trained model f to predict labels for new texts x_{new}. Mathematically: $y = f(x_{new})$ where y_{new} is the predicted label for the new text x_{new}.

After training the model, you can use this function *f* to predict the class of new texts that were not used during training. The model utilizes the acquired knowledge to predict the class of a new text based on its features and performs classification [8].

Let's analyze the test data set based on the training data using the method of Bayesian classification. This model [8] can be used to analyze the mathematical classification of the text. Performs pre-processing of data by tokenizing text xi into words or phrases, and cleans the text of unnecessary information (punctuation, numbers, stop words). We will use the TF-IDF vectorization method to convert text into a numeric vector. This method allows for determining how important and unique words or phrases are for a particular document compared to the entire corpus of texts. The TF-IDF method consists of two components: TF (Term Frequency) – the ratio of the number of occurrences of a certain word (term) in a document to the total number of words in this document. This measures how often a word occurs in a text. The formula for calculating TF can look like this:

$$TF(t,d) = \frac{n_t}{\sum_k n_k}, \tag{5}$$

where n_t – the number of entries of word t in document d;

k is the word index for calculating the total number.

Let's define the inverse document frequency. It is a logarithmically weighted measure of how rare a word is in the entire text corpus. The following formula is used to calculate the IDF [6]*:*

$$IDF(t,D) = \log \frac{|D|}{\left|\{d_i \in D \mid t \in d_i\}\right|}, \tag{6}$$

where $|D|$ is the total number of documents in the corpus collection, and $\left|\{d_i \in D \mid t \in d_i\}\right|$ is the number of documents in the collection D in which the term t occurs (when $n_t \neq 0$). Let's define TF-IDF for a specific term in a document i and calculate it as the product of TF and IDF:

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D). \tag{7}$$

The obtained TF-IDF values are used to create feature vectors [5] for texts in the Naive Bayes classification model, where each value in the vector corresponds to the TF-IDF of each term in the document. These feature vectors can be used for text classification and understanding the importance of individual words in the document. After obtaining a vector representation of the message text, we proceed to the preparation of model data. Let's calculate the a priori probabilities of each class $P(y)$ (probabilities of occurrence of class *y* in the general data set). Calculation of conditional probabilities $P(x,y)$ for each word x_i in the text for each class y. Laplace smoothing was used to avoid zero probabilities:

$$P(x_i, y) = \frac{n_{x_i,y} + 1}{n_y + V}, \tag{8}$$

where $n_{x_i,y}$ is the number of occurrences of word x_i in class y;

n_y is the total number of words in class y;

V is the number of unique words in the training set.

For the new text x_{new}, we calculate probabilities for each class y. Using Bayes' formula, we determine the probabilities of class occurrence:

$$P\left(y \mid x_{new}\right) = P(y) \times \prod_{i=1}^{m} P\left(x_i \mid y\right), \tag{9}$$

where m is the number of words in the text x_{new}.

We choose the class y for which $P(y, x_{new})$ is maximal. This mechanism allows you to perform the classification of new text and predict the class to which an electronic message belongs based on the text data information it has from the training data set. To analyze the classification of electronic messages, a DataSet was collected in which the characteristics of the data were taken into account: the number of examples in the set, the number of features of each example, and the types of these features. The total number of collected and classified messages is 60,000. A target variable and possible classes (spam, not spam) are defined, which can be True or False. The text data for spam and non-spam classification includes information about the number of messages, message categories, their content, and the target variable indicating the class. In this analysis, data preparation was performed for text processing and classification by removing stop words. It is important to note that before performing classification, text data need to be preprocessed, tokenized, and converted into a numerical format (such as word indices or word vectors), and, if necessary, data padding should be applied to equalize the length of the text. Additional data preprocessing, selection of optimal model parameters, and hyperparameter tuning were also performed to achieve better classification results. Alpha Laplace is a smoothing parameter added to the counter of each word to reduce the influence of rare words or variants, and in our case, $\alpha = 0.5$. This parameter was determined using the Grid Search method, which seeks model hyperparameters [6] through cross-validation on a grid of different parameter combinations. Cross-validation is used to evaluate each combination, meaning the data is divided into several folds, the model is trained on some folds and validated on others, and this process is repeated for each parameter combination. This allows to improve the model's efficiency by selecting optimal hyperparameters without manually checking each combination. Some of the used hyperparameters are presented in Table 1.

Table 1 – Table of model hyperparameters

Alpha Laplace	Fit prioritet	Class prioritet
0.1	1	0.2
0.5	1	0.4
1	1	0.4

To balance the classes, the SMOTE upsampling method was used, and this approach was applied only to the training dataset. After processing and analyzing the input data, the initialization and training of the Naive Bayes classifier model for prediction on test data were performed. The result of the work is shown in the graph of Fig. 1.

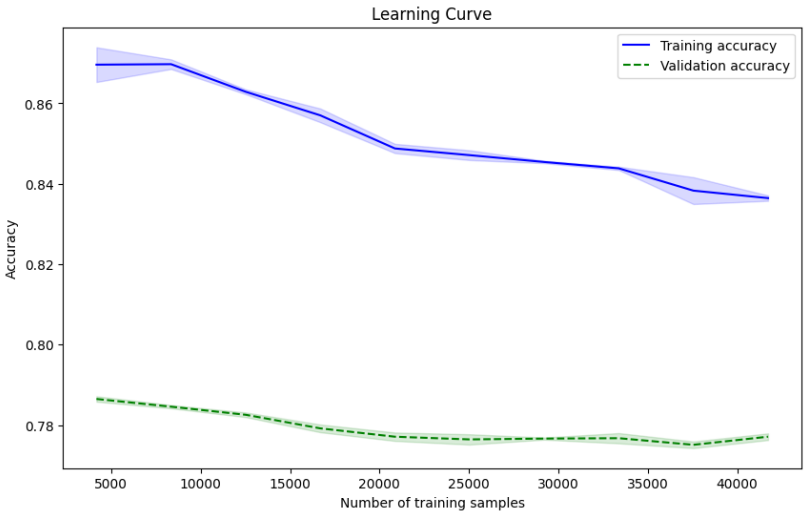

Figure 1 – Learning curve of the Bayesian classification model

The Learning Curve graph allows evaluation of the model's quality based on the number of classified email messages in the training dataset. This graph displays two curves: Training accuracy

and Validation accuracy. The Training accuracy curve shows how well the classification model performs on the data it was trained on. It is expressed as a percentage and represents the ratio of the number of correctly classified instances to the total number of instances in the training dataset. The graph shows that the learning curve reaches 84% accuracy in classifying email messages on the training dataset, which is a satisfactory accuracy indicator. Validation accuracy is a metric that measures the classification model's effectiveness on an independent dataset that the model has not seen during training. This validation dataset is used to evaluate how well the model can generalize its predictions to new data not used during training. This metric achieves 78% accuracy, which can be considered an adequate model to use for classifying electronic messages.

Let's consider the Confusion Matrix - an essential tool for evaluating the results of a classification model. It provides detailed information on how the model classifies data elements, allowing the determination of accuracy levels and errors in prediction. Fig. 2 depicts the Confusion Matrix with quantitative and qualitative indicators.

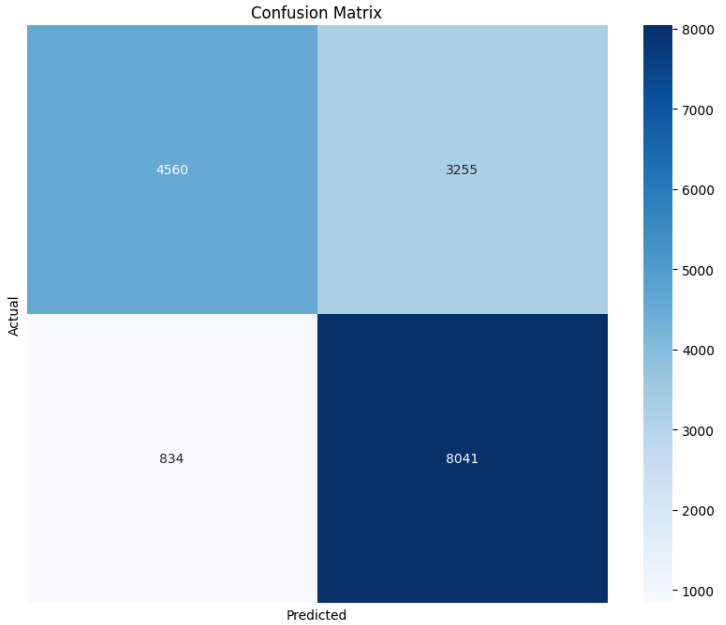

Figure 2 – Confusion matrix of the Bayesian classification model

In this image, “Predicted” represents the predicted class for each message the model is working with. “Actual” is the class (spam or not spam) known at the time of training the model. In the Confusion Matrix, “Actual” shows the actual classification for each message, indicating whether it is spam or not spam, according to the input data. The Confusion Matrix allows visualizing and understanding the real and predicted values of the model. It includes important metrics such as:

– True Positives (TP): The number of correctly classified legitimate messages, $TP_{bc} = 8041$.

– True Negatives (TN): The number of correctly classified spam messages, $TN_{bc} = 4560$.

– False Positives (FP): The number of mistakenly classified legitimate messages, $FP_{bc} = 834$. In this case, the number of non-spam messages that were incorrectly identified as spam.

– False Negatives (FN): The number of mistakenly classified spam messages $FN_{bc} = 3255$, which were incorrectly identified as legitimate.

It's worth noting that this confusion matrix is constructed based on the test dataset to determine the correctness of classifying new messages. The confusion matrix helps determine accuracy, recovery, and other model evaluation metrics. With the help of the matrix, precision, recall, and $F1$ -score can be calculated, providing more detailed information about the model's classification effectiveness. Let's define the precision and recall of the model: $P = 0.783$; $R = 0.782$ and calculate the $F1$-score based on the obtained data: $F1 = 0.782$. The calculation results were obtained using metrics from the Sklearn library.

Let's consider the following classification method based on the Artificial Neural Network (ANN) model. This method of classification of electronic messages is chosen to study the main indicators of classification. In this case, a structure consisting of three Dense layers was chosen: input layer, hidden layer, and output layer. Layer $S1=128$ neurons, $S2=128$, and layer $S3=1$ has one neuron in our case, determined by the type of classification. To prevent overtraining, a Dropout mechanism was added, which provides regularization of neurons by randomly turning off some neurons during training. The value for our model is chosen experimentally as $D=0.3$, which means that 30% of neurons will be randomly turned off. In each layer of the neural network, an activation function is added. The Rectified Linear Activation (RLA) is used for the hidden layer. This function is defined as $F(x)=\max(0,x)$, returning zero for negative values and the same number for values greater than zero, where x is the output signal from the previous input layer of the neural network. According to research, RLA is one of the most popular activation functions. Its use allows the model to learn faster since it performs a simple comparison operation and does not involve computationally complex operations, such as the sigmoid function. In the task of classifying electronic messages, the quick learning ability is an important advantage for the efficient operation of a spam detection system.

Additionally, this activation function helps avoid the vanishing gradient problem that may occur when using other activation functions. The problem of vanishing gradients occurs in neural networks during training and consists of the fact that the gradients that occur during the backpropagation process of determining the class of an electronic message become very small. This means that the weights of neurons start updating too slowly or may not update at all, nearly halting the model training process. The training dataset contains 60 thousand electronic messages. The data has been preprocessed and prepared for an effective model training process. According to the conducted training, the following data were obtained to evaluate the model's quality, as shown in Figure 3.

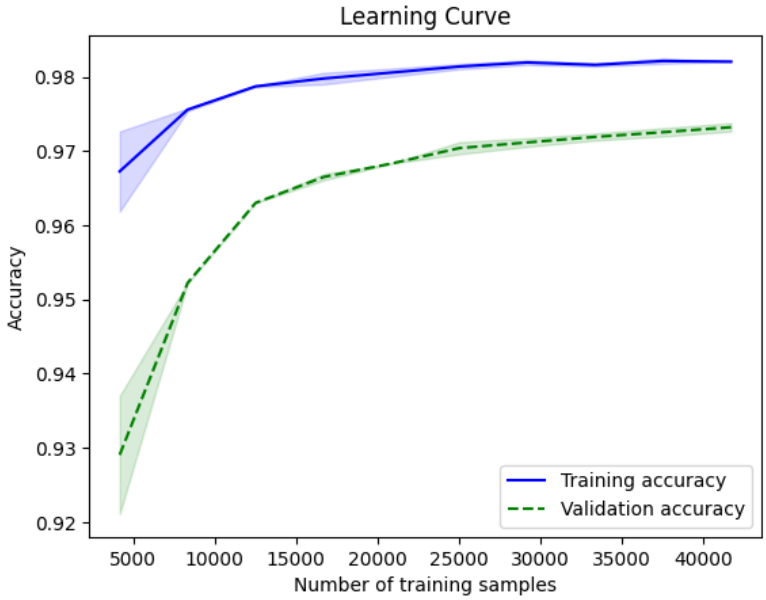

Figure 3 – Learning curve of the Artificial Neural Network model

This graph depicts two curves Training accuracy and Validation accuracy, characterizing the model training process on test and validation data. The Training accuracy curve reflects the quality of classifying the training data. The main goal of the learning curve is to assess how the model's performance, accuracy, and losses change with an increase in the amount of training data. This helps understand whether the model will generalize better to new data with an increase in the volume of training data and evaluate whether it is worth increasing the data volume to improve the model. As evident from the graph, the training metric stabilizes at an accuracy of 98%, indicating a sufficient amount of training data and high accuracy in classifying electronic messages in the training dataset. Using the Validation accuracy metric, we determine the classification effectiveness of the model on

an independent dataset that the model did not analyze during training. This metric achieves an accuracy of 97%, which can be considered adequate for the model's application in classifying electronic messages. It's worth noting an important feature: Training accuracy and Validation accuracy metrics converge and become stable as the data volume increases. This indicates that adding new data helps the model generalize better. Overfitting is absent; the Training accuracy metric demonstrates the model's stable performance, reaching a plateau during training.

Let's consider the confusion matrix for the Artificial Neural Network (ANN) model, with the help of which we will evaluate the result of the classification of electronic messages. The confusion matrix provides detailed information about the model's performance on training and validation data, allowing us to determine the level of accuracy and errors in predictions. This matrix is shown in Figure 4.

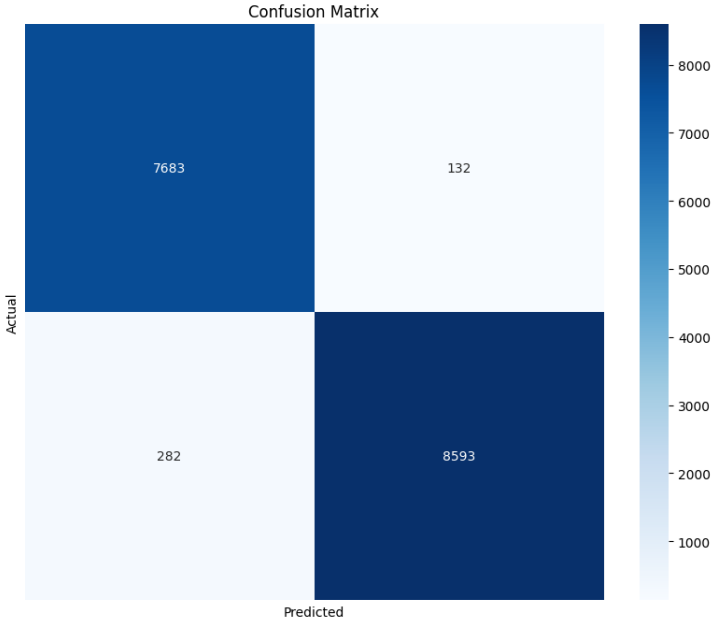

Figure 4 – Confusion matrix of the Artificial Neural Network

Let's define Positive and Negative for each message in this image that the model is working with. According to Figure 4, we obtain the following important metrics:

– True Positives (TP): The number of non-spam emails correctly identified $TP_{ann} = 8593$, as non-spam messages.

– True Negatives (TN): The number correctly identified $TN_{ann} = 7683$, as spam messages.

– False Positives (FP): The number of falsely classified $FP_{ann} = 282$ non-spam emails. In this case, the number of non-spam messages was incorrectly identified as spam.

– False Negatives (FN): The number of falsely classified spam $FN_{ann} = 132$ emails incorrectly identified as non-spam messages.

The data constructed using the confusion matrix was utilized to obtain other metrics. Let's consider the metric Precision: the chosen model determined the proportion of spam emails among all classified messages as spam with precision $P_{ANN} = 0.971$. The accuracy assessment of the model classification using the Recall metric is $R_{ANN} = 0.970$. It determines the proportion of spam emails among all identified as spam that were correctly identified. Having evaluated the accuracy of Precision and Recall metrics, it is possible to consider the $F1$ metric and assess the model's accuracy, which is $F1_{ANN} = 0.971$. Based on the research findings, it is possible to conclude that the accuracy metrics have high values and have been validated using various approaches. Overall performance comparison characteristics and classification accuracy for various methods are provided in Table 2.

Table 2 – Metrics for evaluating the performance of message classification models

	Precision Metric	Recall Metric	*F*1 Metric
Bayesian classifier	0.783	0.782	0,782
Artificial Neural Network	0.971	0.97	0,971

A comparative analysis of the Confusion Matrix based on the selected classification model in Table 3 is an important tool for evaluating the results of multiclass classification, such as message classification. This matrix allows an understanding of the effectiveness of the classification model by analyzing the predictions made by the model and the actual class labels.

Table 3 – Indicators of the quality of classification of electronic messages

	True Positives	True Negatives	False Positives	False Negatives
Bayesian classifier	8041	4560	834	3255
Artificial Neural Network	8593	7683	282	132

By comparing the values of the error matrices, it is possible to draw a conclusion about the accuracy of determining the class of the message and to determine the differences between the Bayesian classifier and Artificial Neural Network models, as well as to estimate the parameters of the models. The overall analysis showed that the Artificial Neural Network (ANN) model outperformed the Bayesian classifier. The ANN model more accurately analyzes and identifies spam and makes fewer errors in determining spam messages.

Conclusions and Future Perspectives. The study of email classification models has led to conclusions regarding their accuracy and completeness in classification. This study allowed for the analysis of confusion matrices and evaluation using Precision, Recall, Accuracy, and F1-score metrics. Through the analysis of models, learning curves were obtained, and the quality of models was assessed using metrics like Training accuracy and Validation accuracy. The evaluation of models on validation data allowed conclusions to be drawn about the accuracy of spam detection and how well the models generalize their knowledge to new data. Data cleaning, missing values, duplicates, and anomalous data processing were performed, contributing to the overall improvement in model quality. Before training the model, the text is tokenized and vectorized. Class balancing was performed to mitigate class imbalance. This approach made it possible to analyze and determine the effectiveness of models, enabling comparisons and drawing conclusions within the context of the email classification task.

Promising directions for further research include the analysis of email classification models using methods and approaches in constructing networks, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN).

REFERENCE

[1] D. Jurafsky, and J. H. Martin, *Speech and Language Processing (2nd ed.)*, London, UK: Pearson Education, 2009.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[3] A. N. Soni, "Spam e-mail detection using advanced deep convolution neural network algorithms", *Journal for innovative development in pharmaceutical and technical science*, vol. 2, iss. 5, pp. 74-80, 2019.

[4] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning", *Decision Support Systems*, vol. 107,

pp. 88-102, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0167923618300010. Accessed on: June 22, 2023.

[5] S. Kumar, A. K. Sharma, and M. Aslam, "A comparative study between naïve Bayes and neural network (MLP) classifier for spam email detection", in *Proc. National Seminar on Recent Advances in Wireless Networks and Communications, NWNC-2014*, vol. 2, 2014. [Online]. Available: https://www.researchgate.net/publication/360426773_A_Comparative_Study_Between_Naive_Bayes_and_Neural_Network_MLP_Classifier_for_Spam_Email_Detection. Accessed on: June 22, 2023.

[6] K. Kowsari, M. K. Jafari, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey", *Information*, vol. 10, iss. 4, 150. (2019). doi: http://dx.doi.org/10.3390/info10040150.

[7] K. U. Santoshi, S. S. Bhavya, Y. B. Sri, and V. Venkateswarlu, "Twitter spam detection using naïve bayes classifier", in *Proc. 2021 6th international conference on inventive computation technologies (ICICT)*, Coimbatore, India, pp. 773-777, 2021. doi: http://dx.doi.org/10.1109/ICICT50816.2021.9358579.

[8] Lv. Teng, Y. Ping, Y. Hongwu, and H. Weimin, "Spam filter based on naive Bayesian classifier", *Journal of Physics: Conference Series*, vol. 1575, no. 1, p. 012054. doi: http://dx.doi.org/10.1088/1742-6596/1575/1/012054.

Стаття надійшла до редакції 30.11.2023.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] D. Jurafsky, and J. H. Martin, *Speech and Language Processing (2nd ed.)*, London, UK: Pearson Education, 2009.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[3] A. N. Soni, "Spam e-mail detection using advanced deep convolution neural network algorithms", *Journal for innovative development in pharmaceutical and technical science*, vol. 2, iss. 5, pp. 74-80, 2019.

[4] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning", *Decision Support Systems*, vol. 107, pp. 88-102, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0167923618300010. Accessed on: June 22, 2023.

[5] S. Kumar, A. K. Sharma, and M. Aslam, "A comparative study between naïve Bayes and neural network (MLP) classifier for spam email detection", in *Proc. National Seminar on Recent Advances in Wireless Networks and Communications, NWNC-2014*, vol. 2, 2014. [Online]. Available: https://www.researchgate.net/publication/360426773_A_Comparative_Study_Between_Naive_Bayes_and_Neural_Network_MLP_Classifier_for_Spam_Email_Detection. Accessed on: June 22, 2023.

[6] K. Kowsari, M. K. Jafari, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey", *Information*, vol. 10, iss. 4, 150. (2019). doi: http://dx.doi.org/10.3390/info10040150.

[7] K. U. Santoshi, S. S. Bhavya, Y. B. Sri, and V. Venkateswarlu, "Twitter spam detection using naïve bayes classifier", in *Proc. 2021 6th international conference on inventive computation technologies (ICICT)*, Coimbatore, India, pp. 773-777, 2021. doi: http://dx.doi.org/10.1109/ICICT50816.2021.9358579.

[8] Lv. Teng, Y. Ping, Y. Hongwu, and H. Weimin, "Spam filter based on naive Bayesian classifier", *Journal of Physics: Conference Series*, vol. 1575, no. 1, p. 012054. doi: http://dx.doi.org/10.1088/1742-6596/1575/1/012054.

ВОЛОДИМИР ОНИЩЕНКО,
АНАТОЛІЙ МІНОЧКІН

АНАЛІЗ МЕТОДІВ КЛАСИФІКАЦІЇ ЕЛЕКТРОННИХ ПОВІДОМЛЕНЬ НА ОСНОВІ МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ

Розглянуто створення механізму виявлення та класифікація повідомлень з оцінкою, наскільки ефективно працюють різні нейронні мережі та можуть розпізнавати, класифікувати різні типи електронних повідомлень, включаючи фішингові атаки, спам, легітимні повідомлення. Виконано попередній аналіз вхідних повідомлень, включаючи їх заголовки, текст та будь-які інші релевантні атрибути. Розглянуто методи підготовки та обробки даних, включаючи векторизацію тексту, видалення шуму та нормалізацію, для використання в навчанні нейронних мереж. Проведена токенізація повідомлення шляхом перетворення на числовий формат з урахуванням виділення ознак. Для текстових повідомлень, важливо виконати токенізацію та векторизацію тексту. Виконано навчання моделі на тестових даних з попереднім розбиттям на дві частини 80% для навчання, 20% для тестування. Навчальний набір використовується для навчання моделі, а тестовий – для оцінки її ефективності. Враховано особливість класової структури даних, а саме рівномірність розподілу класів. В даному випадку спам зустрічається рідше за легітимні повідомлення тому було застосовано техніки балансування класів для забезпечення адекватного навчання моделі. Для балансування класів було обрано техніки випадкове видалення зайвих прикладів, апсемплінг, субдескретизація. Виконана оптимізація параметрів мереж, шляхом дослідження оптимальних параметрів нейронних мереж, такі як кількість шарів, розмір шарів, функції активації, оптимізація гіперпараметрів для досягнення найкращої продуктивності. Оптимізація гіперпараметрів включає визначення оптимальних налаштувань для нейронних мереж, такі як розмір шарів, функції активації, швидкість навчання та інші параметри. Проведена оцінка ефективності шляхом порівняння результатів та продуктивності різних методів класифікації на основі нейронних мереж, використовуючи метрики, такі як точність, відзив, точність та $F1$ -оцінку. Визначино, наскільки методи здатні уникати помилкових класифікацій, коли легітимні повідомлення помилково визнаються спамом, і навпаки. Зроблено порівняння ефективність методів у відношенні до обробки великої кількості повідомлень в реальному часі. На основі аналізу виявлено, наскільки ефективно різні моделі нейронних мереж можуть розпізнавати та класифікувати повідомлення як спам. Розроблено рекомендації на основі результатів аналізу.

Ключові слова: класифікація повідомлень, нейронні мережі, оброблення природньої мови, фільтрація спаму, векторизація тексту, класифікація повідомлень, аналіз тексту, оцінювання якості моделі.

Onishchenko Volodymyr, junior researcher, Institute of special communications and information security National technical university of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine, ORCID 0009-0000-1355-9178, v.o.onishchenko@ukr.net.

Minochkin Anatolii, doctor of technical sciences, professor, leading researcher in Heroiv Krut Military institute of telecommunications and informatization, Kyiv, Ukraine, ORCID 0000-0002-4123-604X, minanatol@gmail.com.

Оніщенко Володимир Олександрович, молодший науковий співробітник, Інститут спеціального зв’язку та захисту інформації Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

Міночкін Анатолій Іванович, доктор технічних наук, професор, провідний науковий співробітник Військового інституту телекомунікацій та інформатизації імені Героїв Крут, Київ, Україна.