Mazurets Oleksandr Viktorovych
Ph.D. in Technical Sciences, Associate Professor,
Associate Professor of Computer Science Department
Khmelnytskyi National University, Ukraine

Sobko Olena Vitalyivna
Teacher of Computer Science Department
Khmelnytskyi National University, Ukraine

Molchanova Maryna Oleksyivna
Teacher of Computer Science Department
Khmelnytskyi National University, Ukraine

Zalutska Olha Oleksandrivna
Graduate student of Computer Science specialty
Khmelnytskyi National University, Ukraine

Yurchak Anton Volodymyrovych
Bachelor student of Computer Science specialty
Khmelnytskyi National University, Ukraine

PRACTICAL IMPLEMENTATION OF NEURAL NETWORK METHOD FOR STRESS FEATURES DETECTION BY SOCIAL INTERNET NETWORKS POSTS

Abstract. *The article considers a neural network method for stress features detection by social internet network posts, designed for automated analysis of text messages posted on social networks in order to identify signs of stress in posts. Based on the designed functional and design architectures of the information system for detecting stress in posts, the software implementation was carried out to study the effectiveness of the developed neural network method for stress features detection by social internet network posts. The practical implementation of the neural network method has determined that the developed method allows detecting stress features in social Internet network posts with an accuracy of 90%.*

Introduction. People spend more and more time on the Internet, especially on social media [1]. This creates new opportunities to study their mental state by analyzing text data, which can be useful for improving their health [2, 3].

In view of recent events in Ukraine, it can be argued that the Internet is gradually becoming a source of threat to the information security of individuals, society and the state [4]. The dissemination of dubious and biased content, together with the use of technologies of information and psychological influence on the minds of individuals, can contribute to increased dissatisfaction with the current government, national conflicts, social tensions and other negative phenomena [5].

Stress is a serious problem in modern society, and detecting it at an early stage can help avoid serious health consequences. Analyzing textual data from social media can help identify signs of stress, even if a person does not explicitly discuss it [6].

In order to detect stress in the messages of social Internet services, it is necessary to perform an analysis of modern theoretical approaches to solving such problems. One of the popular approaches to achieve this goal is to use NLP tools, which allow automated analysis of textual content. NLP is a branch of machine learning that aims to understand and process human language in its various forms. It encompasses both written and spoken language, allowing computers to interact with humans through text messages, audio recordings, speech, and other forms of linguistic expression. NLP is used in a wide range of applications, including automatic language translation, speech recognition, social media sentiment analysis, text generation, spam detection, and more. It opens many opportunities for automation and improvement of communication between computers and people [6].

In the field of NLP, the data used to train models consists of a limited vocabulary. However, there are often non-dictionary terms that are not included in this vocabulary. When processing new texts using a trained model, such terms are usually assigned a generic or standard substitute.

Aim. The aim of the study is to simplify the detection of stressful state in social media posts by analyzing them using NLP tools. To achieve this goal, we have developed a method for detecting stressful state from posts on social media using NLP tools, as well as an appropriate software implementation that will use the created method.

Materials and methods. The method for detecting stress from posts on social media is designed to automate the analysis of text messages posted on social media to detect stress in posts. The scheme and stages of the method are shown in Figure 1.

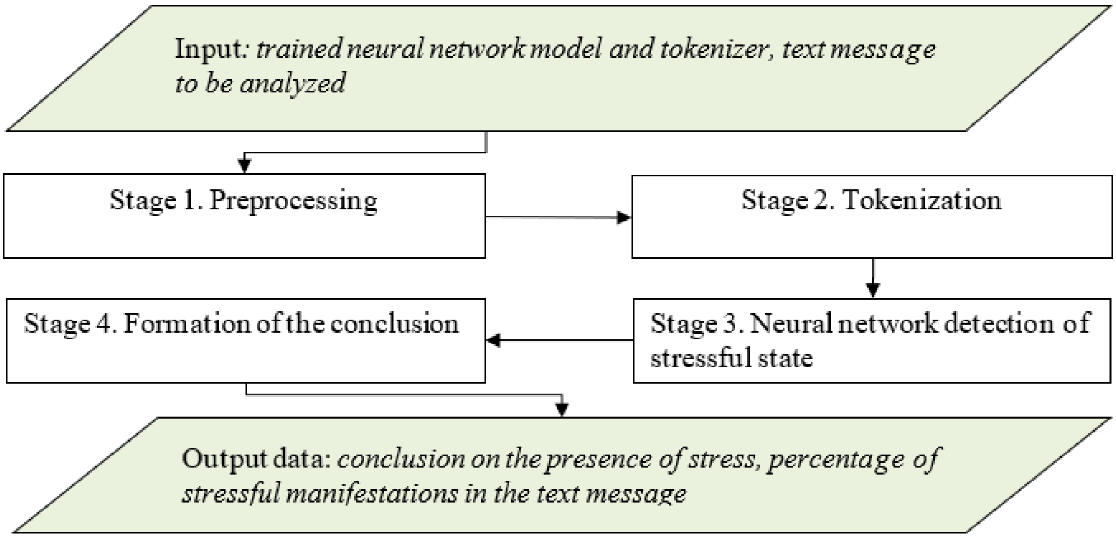

Fig 1. **Scheme and stages of the method for detecting stress by posts**

The input to the method is the trained neural network model and the tokenizer used during training, as well as the text message to be analyzed. The recurrent neural network of the GRU architecture will be used as part of the bachelor's thesis.

The output of the method is a conclusion about the presence of a stressful state, as well as the percentage of stressful manifestations in a text message.

The first stage of the method for detecting stress in posts is preprocessing, which includes removing emoticons, punctuation marks, stop words, etc.

The next stage is tokenization, which is carried out by using the tokenizer loaded in the input data and processing the preprocessed text with it.

The third stage is the neural network detection of a stressful state, which is carried out by processing the loaded trained neural network model of the numerical sequence from the previous stage.

The fourth stage is to form a conclusion about the presence of stress in the text message under analysis, as well as the percentage of its presence.

Since the input data is a trained neural network model, the primary task is to form such models to detect a stressful state. The input data for the formation of the trained neural network model and tokenizer are the dataset and the formed neural network model. The output data are the saved neural network model and the tokenizer.

The first stage is the formation of samples: training and validation. The samples are divided in the ratio of 80 to 20%, where 80% is the training sample and 20% is the validation sample.

The second stage is preprocessing. Both samples (training and validation) undergo the removal of emoticons, punctuation marks, stop words, etc.

The next stage is tokenization. The cleaned training and validation samples undergo the tokenization process, after which the available texts are converted into numerical sequences.

The fourth stage is training the neural network model of the GRU architecture. The model is trained based on the input neural network architecture using the training set. This stage is closely related to the training control stage, which monitors the accuracy and loss function at each training epoch.

The sixth stage is the validation stage, when the trained model tries to classify samples from the validation sample, based on which the neural network model's scores are generated by the Acuracy, Precision, and Recall metrics.

The next step is to save the trained model and the tokenizer if the metrics are above 80%. If the metric values are unsatisfactory, the training process starts over with a change in the neural network architecture in the input data.

The architecture of a neural network is an important aspect of designing an intelligent text analysis information system for detecting signs of stress. As part of the task, we will use a recurrent neural network of the GRU architecture with a dynamic number of layers. The architecture of neural network is shown in Figure 2.

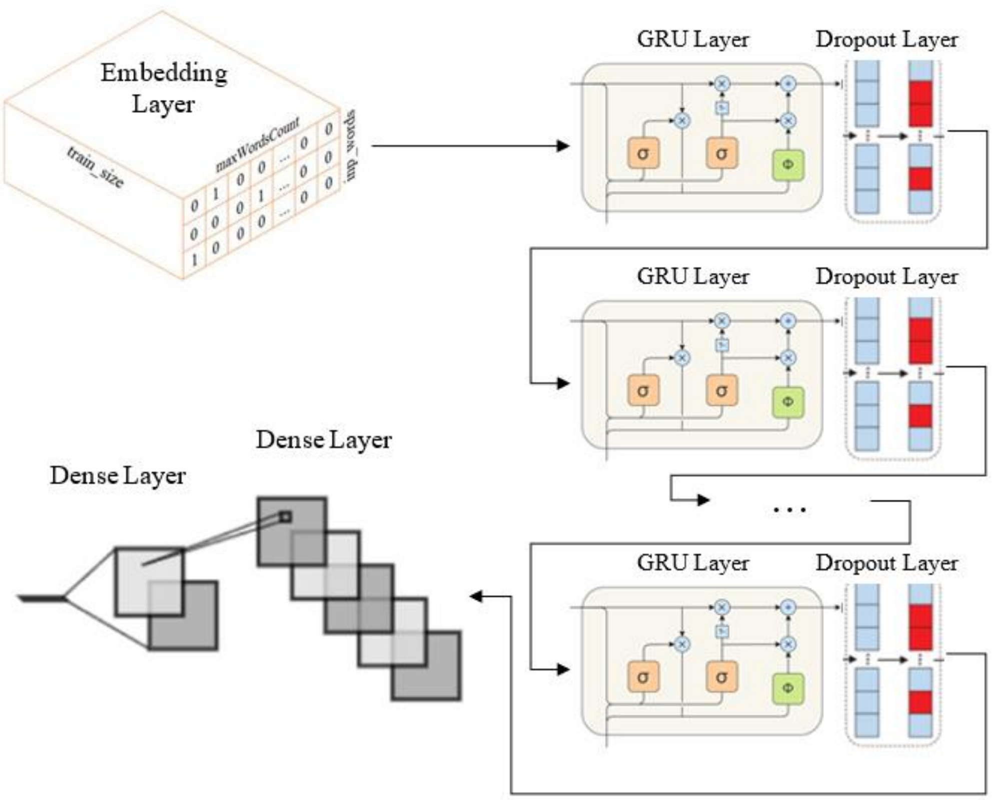

Fig 2. **Neural network architecture**

The required layers of the proposed neural network are the Embedding layer, at least one GRU layer, at least one Dropout layer, and two fully connected Dense layers.

The EmbeddingLayer takes as input the number of words in the dictionary and the dimensionality of the vector representation of each token in the text. Its key task is to convert token sequences into fixed-length vector representations. The main goal of this layer is to learn distributed word representations, so that semantically similar words are located close to each other in space.

The next components of the architecture are the GRU and Dropout layers. GRU is a type of recurrent layer that allows the model to take into account dependencies in sequences and transfer information through time steps, while Dropout prevents overfitting. The number of such connections is determined experimentally depending on the metrics. However, within the framework of the bachelor's qualification work, the number of such components does not exceed five.

The last layers are two fully connected layers (Dense Layer). In the penultimate fully connected layer, the ReLu activation function is used so that the neuron can only transmit positive values. This helps to avoid the problem of gradient vanishing and improves the convergence rate during training.

The last fully connected layer has one neuron and a dynamic activation function (chosen experimentally, but the default is sigmoid).

The functional structure of the information system is logically divided into 3 subsystems and the main menu. The interaction of subsystems is shown in Figure 3.

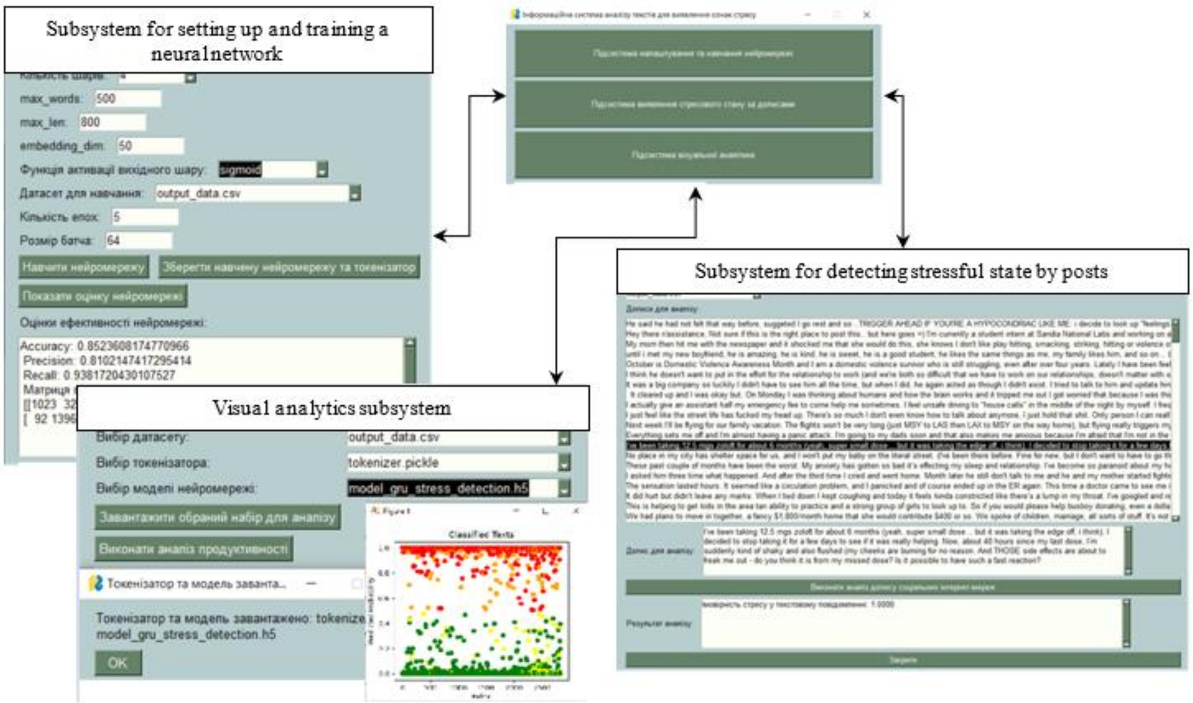

Fig 3. **Interaction of subsystems of the text analysis information system for detecting signs of stress**

The information system for text analysis to detect signs of stress consists of the following subsystems: "Neural Network Tuning and Training Subsystem", "Stress Detection Subsystem by Posts", and "Visual Analytics Subsystem".

The neural network configuration and training subsystem is designed to train and save trained neural network models and their tokenizers according to user parameters. This subsystem performs the following functions:

- entering the number of training epochs;
- entering the batch size;
- selecting a dataset for training and validation samples;
- selecting the number of layers in the architecture;
- entering the dictionary dimension;
- setting the maximum length of the token sequence;
- training the neural network according to user parameters;
- performance analysis based on the metrics of the trained model;
- save the trained neural network model and tokenizer.

The post stress detection subsystem is the main subsystem that uses classifiers trained by the previous subsystem as input. The stress state detection subsystem performs the following functions:

- selecting a dataset for analysis;
- selection of a text message from the selected dataset;
- detailing the selected post for analysis;
- writing a text message manually;
- determining the presence of a stressful state in a text message.

The visual analytics subsystem is an auxiliary subsystem of neural network training, as it allows to expand the functionality of studying the effectiveness of trained classifiers. The functions of this subsystem are:

- selecting a dataset for analysis;
- selecting a tokenizer for analysis;
- selecting a neural network model for analysis;
- downloading them for further verification;
- performing performance analysis.

This subsystem will generate a number of graphs designed to visualize the misclassification of a stressful state.

A diagram of the information system classes is shown in Figure 4.

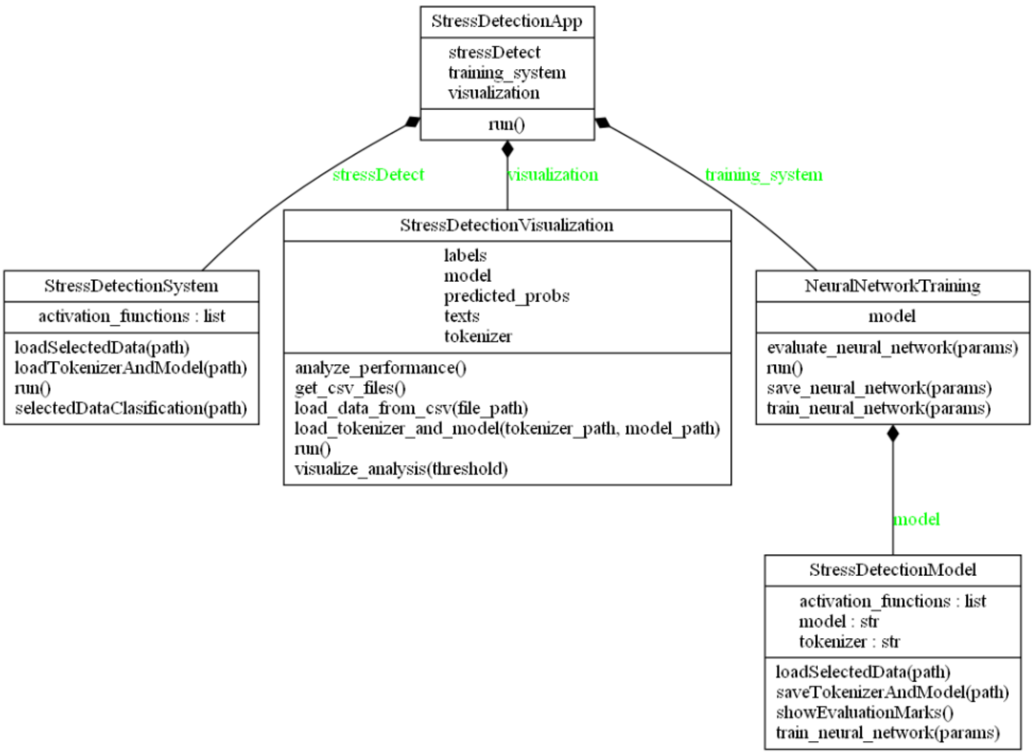

Fig 4. **Class diagram of the text analysis information system for detecting signs of stress**

The NeuralNetworkTraining class is designed to implement a graphical subsystem for training a neural network. It is closely related to the StressDetectionModel class, which contains the algorithmic component for the stages of training the neural network and the tokenizer. The train_neural_network() method implements the logic for training the neural network. The loadSelectedData() method is used to load the training data set from the file system. The saveTokenizerAndModel() method is used to save the trained neural network model and tokenizer to the file system. The showEvaluationMarks() method is designed to display training statistics. In turn, the methods of the NeuralNetworkTraining class, such as train_neural_network(), are intended to be called by clicking on the corresponding control on the user form, and will call the logic of the StressDetectionModel class inside.

The StressDetectionSystem class is designed to implement a subsystem for detecting a stressful state by posts. The loadTokenizerAndModel() method is used to load the model and tokenizer that will be used to detect the stress state. The loadSelectedData() method is used to load data on the user form from the selected dataset. The selectedDataClassification() method is used to detect stress in the message selected by the user.

The StressDetectionVisualization class is designed to implement the visual analytics subsystem. The get_csv_files() method is designed to display all available datasets in the project folder in a drop-down list. The load_tokenizer_and_model method is used to load the selected tokenizer and neural network model for further operation. The visualize_analysis() method is designed to display graphs that demonstrate the correctness of data classification by the neural network.

Results and discussion. According to the designed functional and design architectures of the information system for detecting stressful state by post, we have performed a software implementation that will be key to studying the effectiveness of the developed method for detecting stressful state by posts from social media using NLP tools. The information system, in addition to the usual training parameters such as the number of epochs and the size of the battle, also allows you to change the number of GRU layers, which can vary from 1 to 7 within the software implementation. The GRU layer is used in addition to the Dropout layer to avoid overtraining.

Evaluation of the model for detecting a stressful state using a trained neural network is carried out by loading the newly trained and saved model using the load_model() method. Along with the model, the tokenizer is also loaded, which was used to pre-process the data before feeding it to the neural network.

The data for model evaluation is loaded from a validation file, where each line contains a text entry and the corresponding class label. The text data is converted into sequences using a tokenizer, and then the required length is achieved by adding zeros to the maximum length determined from the output of the first layer of the neural network max_len = model.layers[0].output_dim.

Evaluation is performed by calculating such metrics as accuracy, precision, recall, and confusion matrix using built-in functions from the scikit-learn library. An example of stress state detection is shown in Figure 5.

The visual analytics subsystem is implemented using the Matplotlib library, and its main functionality is to illustrate the possibility of classifying text data with a neural network model. A DataFrame is created that contains information about the text index, predicted probability, true label, and predicted label.

The graphs are divided into two sub-graphs, the first graph shows the texts that were correctly classified. Green color corresponds to correctly classified texts without stress (true label = 0), red color corresponds to correctly classified texts with stress (true label = 1).

The second graph shows the texts that were misclassified. Yellow color corresponds to misclassified texts with stress (true label = 1, but predicted label = 0), orange color corresponds to misclassified texts without stress (true label = 0, but predicted label = 1). The graphs are shown in Figure 6.

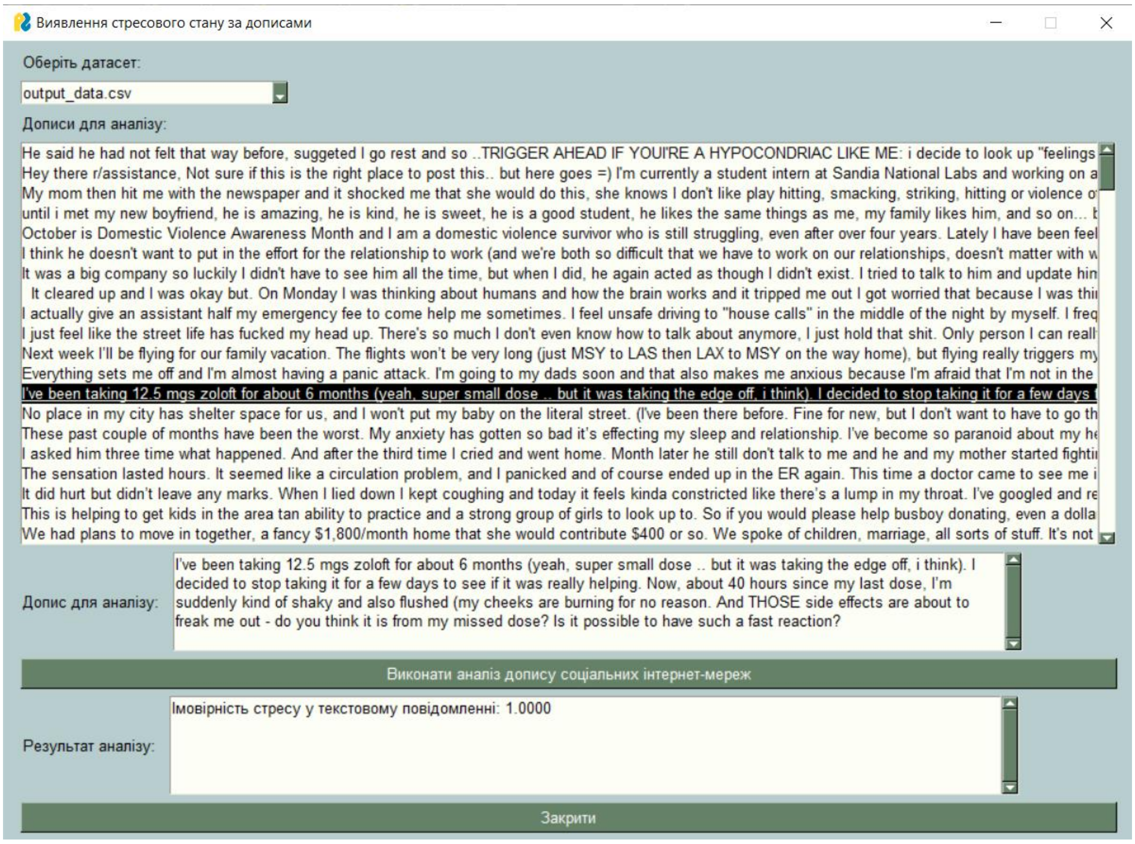

Fig 5. **An example of identifying a stressful state**

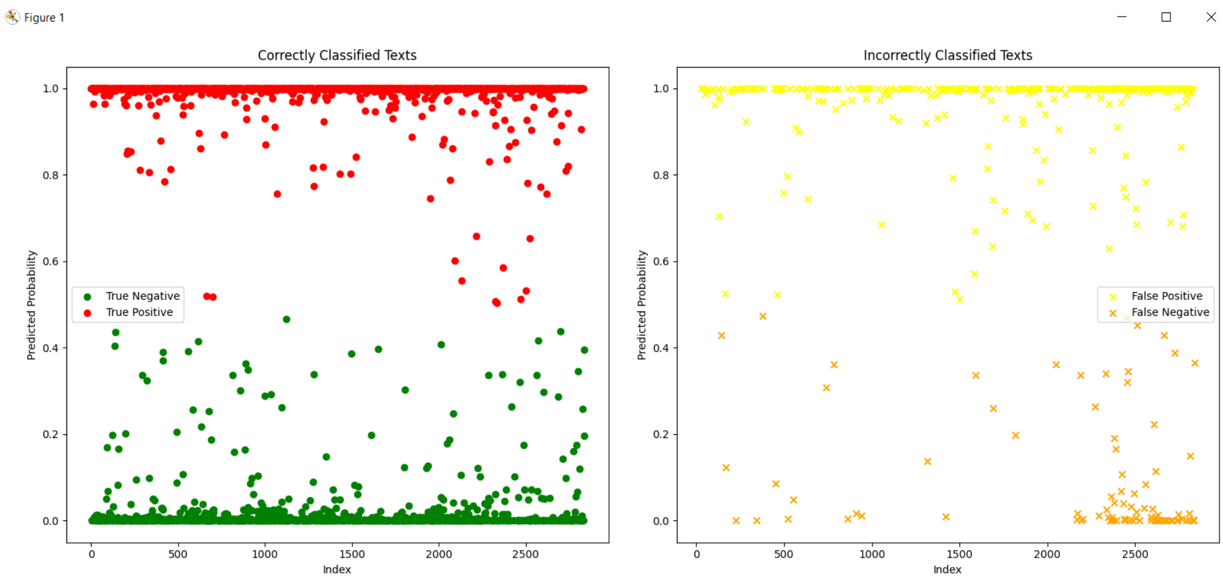

Fig 6. **Illustration of correct and incorrect identification of a stressful state**

The influence of neural network training parameters, such as the number of training epochs, the size of the battle, the number of GRU layers in the architecture, and the influence of tokenizer parameters, was investigated. The neural network models were evaluated using the Accuracy, Precision, Recall, and confusion matrix metrics. Of the options considered, we managed to achieve Accuracy = 0.95, Precision = 0.95, and Recall = 0.94 with batch = 64, epoch = 10, GRU count = 3, and the following tokenizer parameters: max_words = 3000, max_len = 550, and embedding_dim = 300.

Conclusions. Thus, the practical implementation of the neural network method for stress features detection by social media posts was carried out. The purpose of practical implementation is to automate the analysis of text messages posted on social networks in order to detect stress features in social internet network posts.

References:

1. Complete classification of stress: stages and phases of development, types and varieties. DLM. https://dlm.com.ua/povna-klasifikaciya-stresu-stadiï-ta-fazi-rozvitku-vidi-ta-riznovidi.
2. Stress Detection using Machine Learning and Deep Learning. Iopscience. https://iopscience.iop.org/article/10.1088/1742-6596/1997/1/012019/meta.
3. StressPal Frontline. Stresspal. https://stresspal.com/frontline/.
4. Nazarov, N., & Molchanova, M. Information System for Detecting Abusive Speech in Audio Content by Means of Natural Language. In Proceedings of V International Scientific and Practical Conference «Modern strategies of global scientific solutions» (pp. 132–135). International Scientific Unity.
5. Zalutska, O., Molchanova, M., Sobko, O., Mazurets, O., Pasichnyk, O., Barmak, O., & Krak, I. Method for Sentiment Analysis of Ukrainian-Language Reviews in E-Commerce Using RoBERTa Neural Network. In CEUR Workshop Proceedings., 2023, vol. 3387, pp. 344-356.
6. Slobodzian, V., Molchanova, M., Kovalchuk, O., Sobko, O., Mazurets, O., Barmak, O., & Krak, I. (2022). An Approach Based on the Visualization Model for the Ukrainian Web Content Classification. In 12th International Conference on Advanced Computer Information Technologies (ACIT). IEEE. https://doi.org/10.1109/acit54803.2022.9913162