IRSTI 81.96.00 DOI: https://doi.org/10.26577/JMMCS2024121111

Sh.Zh. Mussiraliyeva, M.A. Bolatbek, A.N. Zhumakhanova*, G. Baispay, Zh. Medetbek
Al-Farabi Kazakh National University, Kazakhstan, Almaty
*e-mail: aygerim129@gmail.com

CREATING A MODEL OF SEMANTIC ANALYSIS OF EXTREMIST TEXTS IN THE KAZAKH LANGUAGE

Presently, there is a significant emphasis on the utilization of semantic analysis to scrutinize texts and viewpoints expressed in the Kazakh language within the realm of social networks, with the primary objective of identifying content of a suspicious or extremist nature. This research article is dedicated to exploring the application of machine learning and deep learning techniques in the realm of extremist content detection within textual data.

The investigation takes into account several critical factors, including oversampling and undersampling during the feature processing phase, the nuanced differentiation between extremist and neutral subjects, and the handling of imbalanced classification challenges. These considerations culminate in the development of a sophisticated deep learning model for text classification. The study encompasses the deployment of various machine learning models to discern extremist content within textual materials. Additionally, a comprehensive comparative analysis of machine learning methodologies is conducted to ascertain the most effective approach for this task, taking into consideration oversampling and undersampling techniques for addressing data imbalance issues.

The research endeavors are delineated into two core subtasks: the formulation of a machine learning model specialized in the detection of extremist content within text, and the construction of a deep learning model that factors in the unique characteristics of the Kazakh language and the available dataset.

Furthermore, the study delves into the intricacies of feature processing, culminating in a comparative assessment of outcomes derived from a range of machine learning algorithms used to classify religious extremism, each leveraging distinct feature combinations. The methodologies explored encompass decision trees, random forests, support vector machines, k-nearest neighbors, logistic regression, and naive Bayes.

This research significantly contributes to the spheres of text mining, artificial intelligence, and machine learning, offering practical recommendations for the processing and categorization of texts linked to religious extremism. Moreover, it underscores the contemporary significance of conducting semantic analyses on extremist texts written in the Kazakh language.

Key words: internet extremism, machine learning, deep learning, social networks, neural networks.

Ш.Ж. Мусиралиева, М.А. Болатбек, А.Н. Жумаханова*, Г. Байспай, Ж. Медетбек
Әл-Фараби атындағы Қазақ ұлттық университеті, Қазақстан, Алматы қ.
*e-mail: aygerim129@gmail.com

Қазақ тіліндегі экстремистік мәтіндерді семантикалық талдау моделін құру

Қазіргі уақытта әлеуметтік желілерде қазақ тіліндегі мәтіндер мен көзқарастарды зерттеу үшін семантикалық талдауды қолдануға көп көңіл бөлінуде, оның басты мақсаты күдікті немесе экстремистік сипаттағы мазмұнды анықтау болып табылады. Бұл зерттеу мақаласы мәтіндік деректердегі экстремистік мазмұнды анықтау саласында машиналық оқыту мен терең оқыту әдістерін қолдануды зерттейді.

Зерттеу бірнеше маңызды факторларды ескереді, соның ішінде функцияларды өңдеу сатысында артық іріктеу және жеткіліксіз таңдау, экстремистік және бейтарап субъектілер арасындағы нәзік саралау және теңгерімсіз жіктеу мәселелерін шешу. Бұл пайымдаулар мәтінді жіктеу үшін күрделі терең оқыту моделін әзірлеумен аяқталады. Зерттеу мәтіндік материалдардағы экстремистік мазмұнды анықтау үшін әртүрлі машиналық оқыту үлгілерін пайдалануды қамтиды. Бұдан басқа, деректердің теңгерімсіздігі мәселелерін шешу үшін артық іріктеу және жеткіліксіз таңдау әдістерін ескере отырып, осы тапсырмаға ең тиімді тәсілді анықтау үшін машиналық оқыту әдістемелерінің кешенді салыстырмалы талдауы жүргізіледі.

Зерттеу жұмыстары екі негізгі қосалқы міндетке бөлінген: мәтіндегі экстремистік мазмұнды анықтауға мамандандырылған машиналық оқыту моделін әзірлеу және қазақ тілінің бірегей сипаттамалары мен қолжетімді деректер жиынтығын ескере отырып, терең оқыту моделін құру.

Зерттеу сонымен қатар, діни экстремизмді жіктеу үшін пайдаланылатын машиналық оқыту алгоритмдерінің ауқымынан алынған нәтижелерді салыстырмалы бағалаумен аяқталатын, әрқайсысында ерекше белгілердің комбинациясын пайдалана отырып, мүмкіндіктерді өңдеудің қыр-сырын зерттейді. Зерттелген әдістерге шешім ағаштары, кездейсоқ ормандар, тірек векторлық машиналар, k-ең жақын көршілер, логистикалық регрессия және аңғал Бейс кіреді.

Бұл зерттеу діни экстремизмге қатысты мәтіндерді өңдеу бойынша тәжірибелік нұсқаулар ұсына отырып, мәтінді өңдеу, жасанды интеллект және машиналық оқыту салаларына елеулі үлес қосады. Оның үстіне бұл қазақ тілінде жазылған экстремистік мәтіндерге семантикалық талдау жүргізудің заманауи маңыздылығын көрсетеді.

Түйін сөздер: интернет экстремизмі, машиналық оқыту, терең оқыту, әлеуметтік желілер, нейрондық желілер.

Ш.Ж. Мусиралиева, М.А. Болатбек, А.Н. Жумаханова*, Г. Байспай, Ж. Медетбек
Казахский национальный университет имени аль-Фараби, Казахстан, г. Алматы
*e-mail: aygerim129@gmail.com

Создание модели семантического анализа текстов экстремистской направленности на казахском языке

В настоящее время большое внимание уделяется использованию семантического анализа для изучения текстов и точек зрения, выраженных на казахском языке в социальных сетях, с основной целью выявления контента подозрительного или экстремистского характера. Эта исследовательская статья посвящена изучению применения методов машинного обучения и глубокого обучения в области обнаружения экстремистского контента в текстовых данных.

В исследовании учитывается несколько важных факторов, в том числе избыточная и недостаточная выборка на этапе обработки признаков, тонкая дифференциация между экстремистскими и нейтральными субъектами, а также решение проблем несбалансированной классификации. Эти соображения завершаются разработкой сложной модели глубокого обучения для классификации текста. Исследование включает в себя использование различных моделей машинного обучения для выявления экстремистского содержания в текстовых материалах. Кроме того, проводится всесторонний сравнительный анализ методологий машинного обучения для определения наиболее эффективного подхода к этой задаче с учетом методов передискретизации и недостаточной выборки для решения проблем дисбаланса данных.

Исследовательские усилия разделены на две основные подзадачи: разработка модели машинного обучения, специализирующейся на обнаружении экстремистского контента в тексте, и построение модели глубокого обучения, учитывающей уникальные характеристики казахского языка и доступный набор данных.

Кроме того, исследование углубляется в тонкости обработки признаков, кульминацией которых является сравнительная оценка результатов, полученных с помощью ряда алгоритмов машинного обучения, используемых для классификации религиозного экстремизма, каждый из которых использует отдельные комбинации признаков. Исследованные методологии включают деревья решений, случайные леса, машины опорных векторов, k-ближайших соседей, логистическую регрессию и наивный байесовский подход.

Это исследование вносит значительный вклад в области анализа текста, искусственного интеллекта и машинного обучения, предлагая практические рекомендации по обработке и категоризации текстов, связанных с религиозным экстремизмом. Более того, это подчеркивает современную значимость проведения семантического анализа экстремистских текстов, написанных на казахском языке.

Ключевые слова: интернет экстремизм, машинное обучение, глубокое обучение, социальные сети, нейронные сети.

1 Introduction

Semantic text analysis is one of the main problems of both the theory of creating artificial intelligence systems associated with natural language processing and computer linguistics. Syntactic and morphological analysis is usually used in the primary processing of texts using an automatic machine method. Turning to semantic analysis, scientists not only consider the text as a set of words and sentences, but also try to create a complete semantic image created by the author.

Machine learning is considered a branch of artificial intelligence. Its main idea is that the computer is not limited to using a pre-written algorithm, but learns to solve the problem on its own. Any work can be classified into one of three levels depending on the relative availability of machine learning technology. The first level is when it is available to various technology giants at the level of Google or IBM. The second level is when a student who has certain knowledge can apply it. The third level is when even grandparents can handle it. Now that machine learning is at the intersection of the second and third levels, the pace of changing the world with the help of this technology is increasing every day [1].

To create methods in machine learning, mathematical statistics, numerical methods, mathematical analysis, optimization methods, probability theory, graphical theories, and various methods of working with numerical data are used. As the processing power of computers has increased, the laws and predictions they create have become many times more complex, and the range of problems and problems that can be solved using machine learning has expanded. With machine learning, we use different methods and select the most suitable method for a given task.

The task is divided into 2 subtasks: 1) creating a machine learning model for detecting extremism in text and 2) creating a deep learning model.

In general, many deep learning methods have sufficient performance and can solve many problems that previously could not be solved effectively, for example, they are often used in computer vision, machine translation, and speech recognition tasks. Using deep learning, we develop a model taking into account the characteristics of the Kazakh language and data set [2].

2 Material and methods

Using machine learning to detect extremism in text

In this section, we compare the results of using different machine learning algorithms to classify religious extremism using different combinations of features. Modern research considers the following most common methods for constructing and training classifiers:

decision tree, random forest, support vector machine, k-nearest neighbors, logistic regression, naive Bayes (Figure 1).

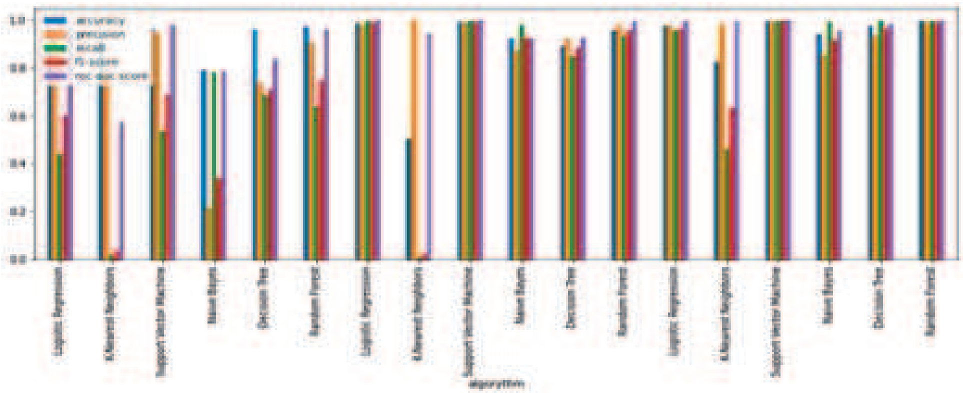


Figure 1: Comparison of results of classification algorithms

Table 1 shows the comparison results between different methods using different features. As shown in the table, the performance of all methods improves by combining more functions together. This observation confirms the informativeness and effectiveness of the acquired features. However, the contribution of each feature varies significantly, indicating variations in the results of individual methods [3].

Table 1: Comparison of different methods using different features

Methods	Features	ACC.	Prec.	Recall	F1	AUC
SVM	Statistical Features +TF-IDF	0.8204	0.2423	0.7593	0.3673	0.8622
	Statistical Features +TF-IDF+POS	0.8412	0.2512	0.6625	0.3643	0.8263
	Statistical Features +TF-IDF+POS+LIWC	0.1065	0.0641	0.8834	0.1196	0.5357
	Statistical Features +TF-IDF	0.9444	0.9529	0.201	0.332	0.6472
Decision tree	Statistical Features +TF-IDF+POS	0.9444	0.8969	0.2159	0.348	0.6395
	Statistical Features +TF-IDF+POS+LIWC	0.9444	0.8812	0.2208	0.3532	0.6274
	Statistical Features	1234	1234	1234	1234	1234
	Statistical Features +TF-IDF	0.9368	1.0	0.0794	0.1471	0.9179
	Statistical Features +TF-IDF+POS+LIWC	0.9364	1.0	0.0744	0.1386	0.914
	Statistical Features +TF-IDF	0.9335	0.8421	0.0397	0.0758	0.5847
KNN	Statistical Features +TF-IDF+POS	0.9354	0.8158	0.0769	0.1406	0.6105
	Statistical Features +TF-IDF+POS+LIWC	0.9351	0.7037	0.0943	0.1663	0.701

	Statistical Features +TF-IDF	0.9681	0.8942	0.6079	0.7238	0.9739
Simplified Bayes algorithm	Statistical Features +TF-IDF+POS	0.9625	0.806	0.598	0.6866	0.9687
	Statistical Features +TF-IDF+POS+LIWC	0.9543	0.7304	0.531	0.6149	0.9599
	Statistical Features +TF-IDF	0.9601	0.9568	0.4392	0.602	0.9759
Logistic regression	Statistical Features +TF-IDF+POS	0.9598	0.9418	0.4417	0.6014	0.9759
	Statistical Features +TF-IDF+POS+LIWC	0.9409	0.6647	0.2804	0.3944	0.9336

The AUC performance measurement in each classification is the area under the receiver operating characteristic curve with all extracted features. In the last column of Table 4, AUC tends to increase with more combined features. The logistic regression method obtains the highest AUC of 0.9759, while most other methods have a very similar AUC value above 0.9.

To evaluate the classification of texts related to extremism with other specific online communities, we expanded our corpus and tested our models in "news" and "jokes" . As the results show, most models show an accuracy of more than 75%, and the model trained with KNN methods detects extremist texts with an accuracy of more than 95% in tests on both datasets. This means our model can work in real environments with an accuracy of about 95% [4].

Using deep learning to detect extremism

Deep learning is ideal for natural language processing (NLP) tasks such as sentiment analysis, text classification, machine translation. For text classification, we use the following deep learning methods: convolutional neural network (CNN) and recurrent neural network LSTM.

Our dataset includes labeled texts. Each text was assigned a label: 0 for neutral text or 1 for extremist text. Since in our case the data is text, we considered filtering and vectorization to prepare the data. Text filtering is performed to reduce noise and outliers.

The following algorithm was used to filter the texts:

1) bringing all characters to the same case and removing unnecessary characters,

2) exclusion of common words (stop words),

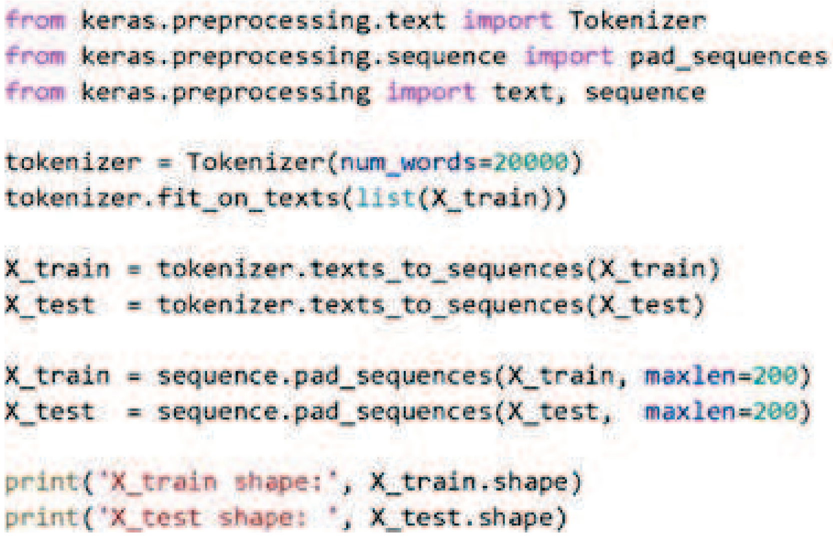
3) perform stemming and lemmatization,

4) indicate the tokenization pattern (breaking the text into words - tokens) and the n-gram model of words (the number of possible words in a token).

Neural networks can only learn to find patterns in numeric data, and so before we feed text into the neural network as input, we converted each word into a numeric value. This process is called word encoding or tokenization.

For tokenization, we used word embeddings. This method represents words as dense word vectors (also called word embeddings). This means that the word "embedding" collects more information into fewer dimensions. Their goal is to map semantic meaning into geometric space. This geometric space is called the embedding space. This will display semantically similar words close to the embedding space, such as numbers or colors.

To tokenize the data, we used the Tokenizer utility class in Keras, which can vectorize a text corpus into a list of integers.



```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing import text, sequence

tokenizer = Tokenizer(num_words=20000)
tokenizer.fit_on_texts(list(X_train))

X_train = tokenizer.texts_to_sequences(X_train)
X_test  = tokenizer.texts_to_sequences(X_test)

X_train = sequence.pad_sequences(X_train, maxlen=200)
X_test  = sequence.pad_sequences(X_test,  maxlen=200)

print('X_train shape:', X_train.shape)
print('X_test shape: ', X_test.shape)
```

In Tokenizer we used two parameters: num_words which is responsible for setting the size of the dictionary and pad_sequence() which simply pads the sequence of words with zeros. The pad_sequence() parameter is used to solve the problem of different word lengths in a text sequence. We add the num_words parameter, which is responsible for setting the size of the dictionary. We set the value of num_words to 20000. One of our problems is that each text sequence in most cases has a different word length. And we'll also add the maxlen parameter to indicate how long the sequences should be. This cuts out sequences larger than this number [5-6].

3 Literature review

The objective of this literature review is to bridge an existing knowledge gap by conducting a comparative examination of machine learning algorithms utilized for the identification of extremist content in the Kazakh language. By amalgamating and critically assessing prior research, this review seeks to provide insights into the strengths, limitations, and potential avenues for future exploration within this field. Ultimately, this endeavor contributes to the advancement of robust tools aimed at countering the proliferation of extremism within the Kazakh online sphere.

Article [7] underscores the importance of identifying and categorizing tweets associated with extremism, as extremist groups employ social media platforms to disseminate their ideologies and recruit adherents. The article introduces a system designed to analyze content related to terrorism, with a specific focus on classifying tweets into extremist and non-extremist categories. This approach harnesses deep learning-based sentiment analysis techniques to construct a tweet classification system. Encouragingly, the experimental results suggest the potential effectiveness of this structural framework. The article addresses the pressing need for effective methodologies to detect extremist content on prominent social networks like Facebook and Twitter. It makes a substantial contribution to the domain of extremism studies by presenting a framework that aids in monitoring and combating the propagation of extremist ideologies online. Furthermore, this work offers prospective researchers a blueprint for advancing the field of identifying and categorizing extremist content on social media platforms.

In Article [8], the significance of researching online extremism to monitor the proliferation of hate on social media platforms is articulated. The author highlights the limitations of existing research, emphasizing its ideological bias and its propensity to utilize simplistic binary or tertiary classifications. The research within this article endeavors to establish a balanced dataset encompassing various ideologies, with a particular focus on extremist tweets. The resulting dataset, referred to as Merged ISIS/Jihadist-White Supremacist (MIWS), is evaluated employing pretrained BERT and its variants (RoBERTa and DistilBERT), achieving a notable f1 score of 0.72. This study underscores the increasing emphasis on natural language processing employing deep learning techniques within extremism detection research.

Article [9] delves into the role of uncertainty in political, religious, and social matters in inciting extremism among individuals, which manifests through their expressions on social networks. Acknowledging the dominance of English in social media interactions, this research underscores the importance of considering sentiments expressed in other local languages to gain a more comprehensive understanding of the data. The study concentrates on sentiment analysis of multilingual textual data sourced from social networks to gauge the intensity of extremist sentiment. It introduces a multilingual dictionary complete with intensity weightings, achieving a validation accuracy of 88%. For classification, Polynomial Naive Bayes and Linear Support Vector Classifiers are deployed, with the Linear Support Vector Classifier attaining an 82% accuracy rate on a multilingual dataset. This research advances our comprehension of extremist sentiments expressed in multiple languages on social networks, offers insights into the levels of extremism, and underscores the effectiveness of the classification algorithms employed.

The subsequent article [10] accentuates the menace posed by online extremists on social media platforms and acknowledges the limitations associated with suspending their accounts, as they can readily create new ones. This study proffers operational solutions to confront this challenge, with a particular focus on formulating behavioral patterns for Twitter accounts linked to the "Islamic State of Iraq and Syria" (ISIS). These patterns are employed to track existing extremist users by identifying pairs of accounts attributed to the same individual.

In summation, these articles collectively enrich the landscape of detecting extremist content through the application of machine learning and deep learning techniques. They encompass diverse facets such as sentiment analysis, language models, social network analysis, and deep learning architectures. By engaging with these articles, readers can acquire a comprehensive comprehension of the subject matter, along with insights into the diverse methodologies and algorithms employed in this domain.

4 Results and discussion

In neural network, we know several terms such as input layer, hidden layer and output layer. Thus, the difference between deep learning and neural network architecture is the specified number of hidden layers. A simple neural network has only 1 hidden layer, whereas Deep Learning has more than 1 hidden layer (Figure 2).

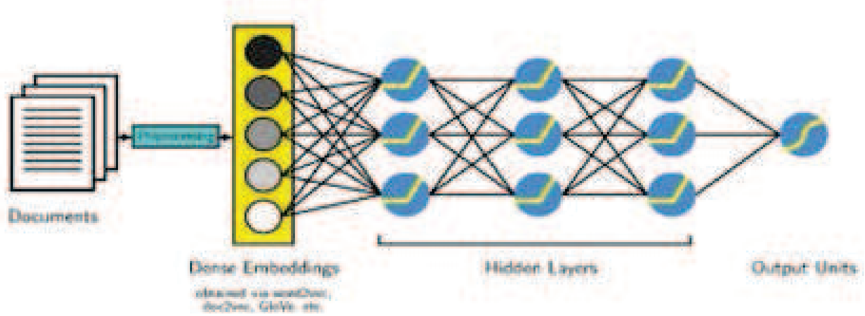


Figure 2: Deep Learning Model Architecture

We start with a layer of input neurons, where we enter our feature vectors, and then the values are transferred to the hidden layer. Each time we connect, we pass the value forward while the value is multiplied by the weight and the offset is added to the value. This happens on every connection, and at the end we get the value of the output layer. The output layer consists of one or more output nodes. In our case, one node, since we have a binary classification task.

Neural network formula: To calculate the values for each output node, we must multiply each input node p by the weight W and add a bias b. All this must then be summed up and passed to function f. This function is considered an activation function, and there are various functions. Typically, a rectified linear unit (ReLU) is used for hidden layers, a sigmoid function for the output layer in a binary classification problem, or a softmax function for the output layer in multiclass classification problems. We use the sigmoid function. The algorithm starts by initializing the weights with random values and then trains them using a method called backpropagation. This is done using optimization techniques (also called an optimizer), such as gradient descent, to reduce the error between the calculated output and the desired output (also called the target output). The error is determined by the loss function, the losses of which we want to minimize using the optimizer. Here we use the "Adam" optimizer and the "cross entropy" loss function [11].

Development of a convolutional neural network for text classification

Convolutional neural networks have revolutionized image classification and computer vision by being able to extract features from images and use them in neural networks. The properties that make them useful for image processing also make them useful for sequence processing. In the case of text classification using CNN, the convolutional kernel slides over word embeddings, only its task is to look at embeddings for several words at once. The dimensions of the convolutional kernel must also change to suit this task.

To look at word embedding sequences, we want the window to look at multiple (usually 3 or 5) word embeddings in the sequences. The cores will be a wide rectangle with dimensions like 3x300 or 5x300 (with an embedding length of 300). In our case 4x200 because we set the sequence length to 200 when tokenizing. Each kernel cell has a corresponding weight. As the kernel slides over the word embedding, the kernel's weights are multiplied by the value of the word embedding, then all the multiplied values are summed to produce the output value.

The convolutional neural network will include many of these kernels, and as the network is trained, these kernel weights are learned. Each core is designed to view a word and surrounding words in a sequential window. Thus, the convolution operation can be considered as window-based feature extraction. There is another nice property of this convolution operation. Recall that similar words will have similar embeddings, and the convolution operation is simply a linear operation on these vectors. So, when a convolutional kernel is applied to different sets of similar words, it will produce the same output value [12].

To process the entire sequence of words, these kernels will sequentially traverse the list of word embeddings. This is called 1D Convolution because the kernel only moves in one dimension: time. One core will move one by one through the list of input embeddings, looking at the first word embedding, then the next word embedding, the next, and so on. The resulting output will be a feature vector.

The maximum values obtained by processing each of our convolutional feature vectors will be concatenated and passed to the last layer. This is called MaxPooling. And this is what our convolutional neural network looks like (Figure 3).

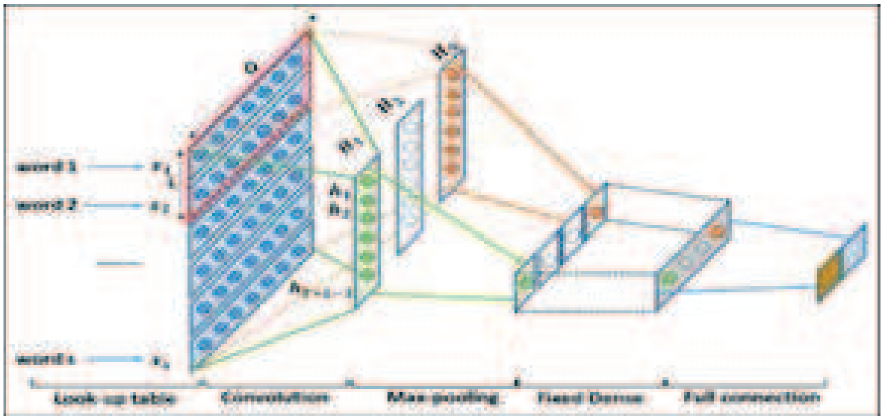
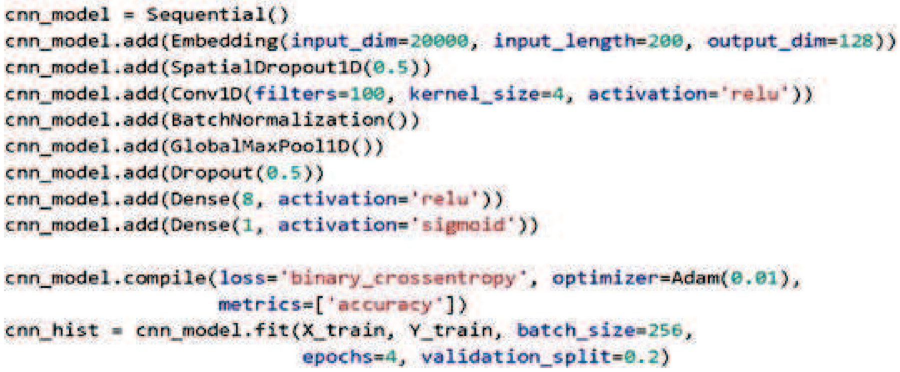


Figure 3: Proposed Convolutional Neural Network

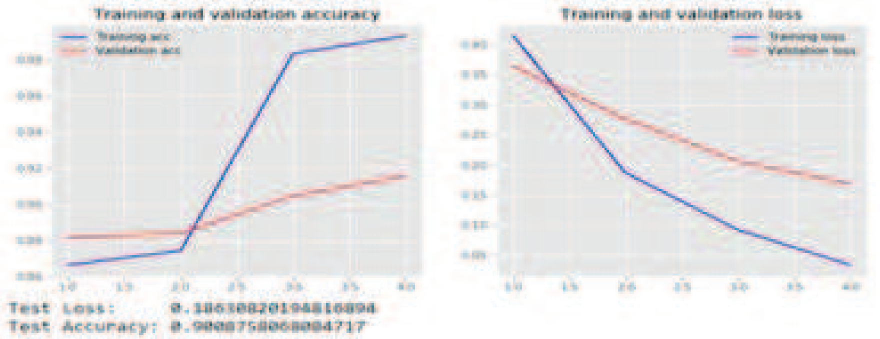
Now let's see how we can use this network in Keras. First, we need to add an embedding layer with the parameters input_dim - the size of the dictionary, the number of unique words we want to use; input_length – sequence length; output_dim – dimension of the embedded variable. We then set an exclusion layer to exclude 50% of the nodes. Now we add a convolutional layer that has 100 filters with a kernel size of 4, so that each convolution takes into account a window of 4 word embeddings and a relu activation function. Before we add the Max Pooling layer, we add a normalization layer. After the pooling layer, we add a dense layer to get a pin size of 8 and use the relu activation function. Finally, we set up the output layer. Since we are doing binary classification, we use the sigmoid activation function and get 1 result in the output layer.

Here we used the "Adam" optimizer and the "cross entropy" loss function. And we got the following result for 4 epochs (Figure 4) [13]:



```
cnn_model = Sequential()
cnn_model.add(Embedding(input_dim=20000, input_length=200, output_dim=128))
cnn_model.add(SpatialDropout1D(0.5))
cnn_model.add(Conv1D(filters=100, kernel_size=4, activation='relu'))
cnn_model.add(BatchNormalization())
cnn_model.add(GlobalMaxPool1D())
cnn_model.add(Dropout(0.5))
cnn_model.add(Dense(8, activation='relu'))
cnn_model.add(Dense(1, activation='sigmoid'))

cnn_model.compile(loss='binary_crossentropy', optimizer=Adam(0.01),
                  metrics=['accuracy'])
cnn_hist = cnn_model.fit(X_train, Y_train, batch_size=256,
                         epochs=4, validation_split=0.2)
```



```
Test Loss:     0.18630820194816894
Test Accuracy: 0.9008758068084717
```

Figure 4: Result for 4 epochs

Application of a recurrent neural network

A recurrent neural network is a deep learning algorithm designed to solve a variety of complex computer problems, such as object classification and speech detection. RNNs are designed to process a sequence of events that occur sequentially, making sense of each event based on information from previous events. RNNs are rarely used in real-world scenarios due to the vanishing gradient problem. This is one of the biggest challenges for RNN performance. In practice, the RNN architecture limits its long-term memory capabilities, which are limited to remembering only a few sequences at a time.

LSTM (Long short-term memory) is designed to solve the vanishing gradient problem and allow them to retain information for longer periods of time compared to traditional RNNs. Therefore, we use LSTM rather than a traditional recurrent neural network. The LSTM architecture is shown below (Figure 5).

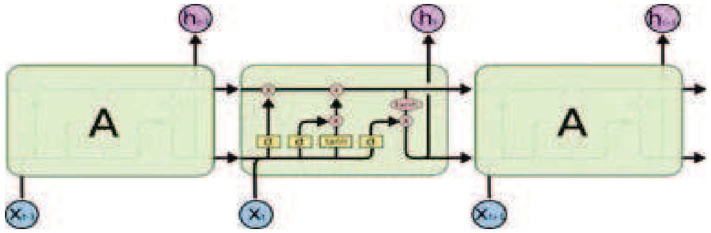


Figure 5: LSTM architecture

We will classify text data using a deep learning network with long short-term memory (LSTM). Text data is naturally sequential. A piece of text is a sequence of words between which there may be dependencies. To learn and use long-term dependencies to classify sequence data, we use an LSTM neural network. An LSTM network is a type of recurrent neural network (RNN) that can learn long-term dependencies between time steps, as shown above, of sequence data.

To input text into an LSTM network, you first need to convert the text data into numeric sequences. We already converted the text into numeric values when we built the convolutional neural network model. Next we will work with those numerical values.

We created exactly the same model for this neural network as for the convolutional neural network. Only instead of a convolutional layer, a bidirectional LSTM layer was added.

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance in sequence classification tasks. In problems where all time slots of the input sequence are available, bidirectional LSTMs train two LSTMs instead of one in the input sequence. The first one refers to the input sequence as is, and the second one refers to an inverted copy of the input sequence. With this form of generative deep learning, the output layer can simultaneously receive information from the backward and forward states (Figure 6).

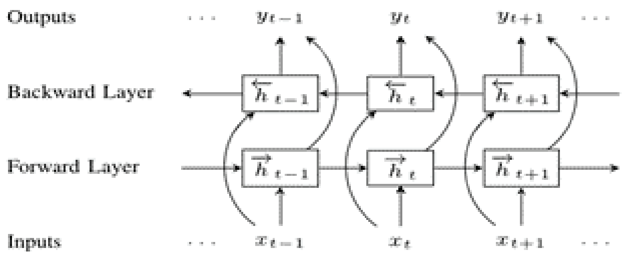


Figure 6: LSTM architecture

A convolutional neural network (CNN) is limited by the local window size and can only extract local text features. For long texts such as news, CNN cannot learn the long-term dependency of long text. A deep learning recurrent neural network model based on long short-term memory (LSTM) can learn the long-term dependency of text. The test data classification results are shown in Figure 7 [14].

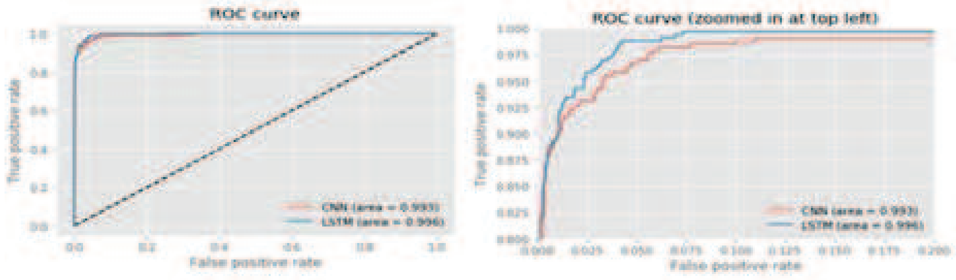


Figure 7: Test data classification result

5 Conclusion

As a result of this article, the following results were obtained: a) different machine learning models were applied to the task of detecting extremism in text content; b) a comparative analysis of machine learning methods was carried out to select the optimal method for a given task; c) oversampling and undersampling methods were carried out to eliminate the problem of data imbalance; d) a deep learning model (convolutional and recurrent neural networks) was developed to detect extremism in Kazakh texts.

Funding Statement. This research was carried out within the framework of the project "Multi-classification of ideological directions of cyber extremism in the Kazakh language using artificial intelligence methods" , funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant No. AP19676342, project leader Sh. Zh. Mussiraliyeva).

References

[1] Bolatbek M.A., Mussiraliyeva Sh.Zh., "Identification of extremist texts using machine learning methods" , *Bulletin of KazUTZU* 6 (130) (2018): 300–304.

[2] Yntykbai B.N., Mussiraliyeva Sh.Zh., Bolatbek M.A., "Analysis of security and confidentiality in social networks using machine learning methods" , *Materials of the International Scientific Conference of Students and Young Students "Farabi World"* Almaty: Kazakh University, (2021): 119.

[3] Chesnokov V.O., "The application of the algorithm of selection of communities in information warfare in social networks" , *Questions of cyber security,* 1 (19) (2017): 37–44.

[4] Ripeanu, Beznosov K., Santos-Neto E., "Thwarting fake OSN accounts by predicting their victims" , *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security,* (2015): 81–89.

[5] Basu A., "Social network analysis: A methodology for studying terrorism" , *Social Networking, ser. Intelligent Systems Reference Library,* 65 (2014): 215–242.

[6] Freeman M., "The Sources of Terrorist Financing: Theory and Typology" , *Studies in Conflict & Terrorism,* 34 (2011): 461–475. Doi:10.1080/1057610X.2011.571193.

[7] Ahmad S., Asghar M.Z., Alotaibi F.M., Awan I., "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques" , *Human- centric Computing and Information Sciences,* 9 (24) (2019): 1–23. Q1.

[8] Mayur G., Swati A., Ketan K., Ajith A., "Multi-ideology Multi-class Extremism Classification using Deep Learning Techniques" , *IEEE Access,* (2022). Q1.

[9] Asif M., Ishtiaq A., Ahmad H., Aljuaid H., Shah J., "Sentiment analysis of extremism in social media from textual information" , *Telematics Informat.,* 48 (2020): 101345. Q1.

[10] Klausen J., Marks C.E., Zaman T., "Finding extremists in online social networks" , *European Journal of Operational Research,* 66 (4) (2018): 957–976. Q1.

[11] Taha K., Yoo PD., "Shortlisting the influential members of criminal organizations and identifying their important communication channels" , *IEEE Transactions on Information Forensics and Security,* 14 (8) (2019): 1988–1999.

[12] Devyatkin D.A., Smirnov I.V., Ananyeva M.I., Kobozeva M.V., Chepovskiy A.M., Solovyev F.N., "Exploring linguistic features for extremist texts detection (on the material of Russian-speaking illegal texts)" , *IEEE International Conference on Intelligence and Security Informatics (ISI),* (2017): 188–190.

[13] Bissaliyev M.S., Nyussupov A.T., Mussiraliyeva Sh.Zh., "Enterprise Security Assessment Framework for Cryptocurrency Mining Based on Monero" , *Vestnik KazNU Series "Mathematics, Mechanics, Informatics",* 2 (98) (2018): 67–76.

[14] Nouh M., Nurse J., "Identifying Key Players in Online Activist Groups on Facebook Social Network" , *IEEE Computer Society,* (2015): 969–978.