Computing, Telecommunication and Control, 2022, Vol. 15, No. 4, Pp. 98–107.
Информатика, телекоммуникации и управление. 2022. Том 15, № 4. С. 98–107.

Research article
DOI: https://doi.org/10.18721/JCSTCS.15408
UDC 004.8

ONE-DIMENSIONAL CONVOLUTIONAL LAYERS IN A NEURAL NETWORK FOR WIND SPEED TIME SERIES ANALYSIS

D.N. Kobzarenko[1] ✉ , *A.G. Mustafaev*[2], *Z.A. Gasanova*[3], *D.S. Magomedova*[4]

[1,2,3,4] Dagestan State University of National Economy, Makhachkala, Russian Federation

✉ kobzarenko_dm@mail.ru

Abstract. Data analysis using neural networks and deep machine learning is one of the current trends in scientific research in various fields. One of the scientific tasks of this direction is the study and prediction of time series using artificial intelligence. The article discusses the results of experiments on adding one-dimensional convolutional layers to a neural network within the framework of the task of classifying meteorological time series data – wind speed. The accuracy of the forecast is shown to increase due to the inclusion of one-dimensional convolutional layers in the model. The increase in accuracy on the test data set for the problem under consideration is about 9.5 %. Several variants of architectures for building a model with one-dimensional convolutional layers and evaluating the accuracy of their classification after machine learning are given. The results obtained allow us to conclude that the use of one-dimensional convolutional layers in the neural network architecture is effective for identifying and predicting a time series of meteorological parameters.

Keywords: neural network, deep machine learning, 1D convolutional layer, wind speed, time series

Citation: Kobzarenko D.N., Mustafaev A.G., Gasanova Z.A., Magomedova D.S. One-dimensional convolutional layers in a neural network for wind speed time series analysis. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 98–107. DOI: 10.18721/JCSTCS.15408

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.15408
УДК 004.8

ОДНОМЕРНЫЕ СВЕРТОЧНЫЕ СЛОИ В НЕЙРОННОЙ СЕТИ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ – СКОРОСТЕЙ ВЕТРА

Д.Н. Кобзаренко[1] ✉ , *А.Г. Мустафаев*[2], *З.А. Гасанова*[3], *Д.С. Магомедова*[4]

[1,2,3,4] Дагестанский государственный университет народного хозяйства, г. Махачкала, Российская Федерация

✉ kobzarenko_dm@mail.ru

Аннотация. Анализ данных с использованием нейронных сетей и глубокого машинного обучения является одним из современных трендов в научных исследованиях в различных областях. Одна из научных задач этого направления – исследование и прогнозирование временных рядов с помощью искусственного интеллекта. В статье рассмотрены результаты экспериментов по добавлению одномерных сверточных слоев в нейронную сеть в рамках задачи классификации данных метеорологических временных рядов – скоростей ветра. Показано повышение точности прогноза за счет включения в модель одномерных сверточных слоев. Повышение точности на наборе тестовых данных для рассматриваемой задачи составляет около 9,5 %. Приведены несколько вариантов архитектур для построения модели с одномерными сверточными слоями и оценка точности их классификации после машинного обучения. Полученные результаты позволяют сделать вывод об эффективности применения одномерных сверточных слоев в архитектуре нейронной сети для идентификации и прогнозирования временного ряда метеорологических параметров.

Ключевые слова: нейронная сеть, глубокое машинное обучение, 1D сверточный слой, скорость ветра, временные ряды

Для цитирования: Kobzarenko D.N., Mustafaev A.G., Gasanova Z.A., Magomedova D.S. One-dimensional convolutional layers in a neural network for wind speed time series analysis // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 98–107. DOI: 10.18721/JCSTCS.15408

Introduction

In modern scientific researches and developments, artificial intelligence, mostly represented by neural network models, occupies its niche as a tool for solving a group of tasks, including wind energy parameters prediction [1–3]. As a rule, these tasks are regression, forecasting, classification, generating new data based on templates, searching for outliers in data, etc. At the same time, neural network models can be used not only to solve tasks with a classical formulation, where there are initial data and it is required to obtain the final result with the required accuracy, but also to carry out scientific researches that provide answers to questions about the properties of data arrays.

The relevance of research on wind monitoring data is determined by the high interest in the development of renewable energy sources. The main wind power characteristic is the speed. But the wind direction is also an important parameter, since it affects the orientation of the wind turbine and its efficiency during rotation.

To implement the research, we prepared a database based on observations from four meteorological stations located on the territory of the Republic Dagestan (Russian Federation) for the time period of 2011–2020. The names of the stations, "Akhty", "Derbent", "Kochubey", and "Makhachkala", coincide with the names of the corresponding settlements in the region.

Analysis of wind speed and direction time series using neural network models is a continuation of work [4]. The work showed seasonal changes in the frequency characteristics of the meteorological time series in the Coastal Dagestan.

In paper [5], the analysis of meteorological time series (wind speed and direction) was performed using neural network models for the classification task built on the basis of fully connected layers only. This research showed that wind direction time series are classified with almost 100 % accuracy. However, for wind speed time series, the situation is different: in this case, the overall classification accuracy does not reach 70 %. This suggests that the wind speed time series related to the territories of the region that are close in distance contain much less obvious patterns that cannot be captured by a network built only based on fully connected layers.

Use of one-dimensional convolutional layers [6] is a way to improve the performance of a neural network model. One-dimensional convolutional layers are used in a wide variety of tasks from different science areas and technologies [7–12]. Therefore, this paper discusses the results of experiments on use of one-dimensional convolutional layers for the wind speed time series classifying task.

Why classification but not regression? We suppose that performing classification before regression allows us to answer the following questions:

– How well does a neural network recognize a meteorological station by wind parameters?
– What recognition is better: by wind speed data or by wind direction data?
– What data time intervals are optimal for recognition?
– Which meteorological stations are recognized better or worse?
– How are recognition errors correlated with the geographical and relative location of a meteorological station?

The information obtained as a result of the classification task is very important in performing time series prediction based on the same neural network. It significantly expands knowledge about the object of survey – wind speed and wind direction.

We found no publications considering the meteorological time series classifying task in the same way and knew with certainty that such researches had not been performed for our region before. Therefore, we started from our own results.

Toolkit and data preparation

The work used one of the most popular modern tools – the *Keras* library for the *Python* programming language. *Keras* is a high-level add-on to the basic *Tensorflow* neural network library, which allows you to create models on *Python* more quickly than for example *PyTorch* or *Tensorflow* (separately from *Keras*). As a programming environment, the Google Colaboratory cloud resource was used: it provided the ability to do all the work, display and save the results in an Internet browser.

The source data were a set of text files with structured meteorological data. Each text file contained data observations for a month. The frequency of observations was eight times a day in uniform time intervals (hours:minutes): 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00.

The data covered the time period of 2011–2020. Each meteorological station was represented by a set of 120 files containing information for 10 years. The first engineering challenge was to combine all the data into a single table. A *Python* script was written to solve it. As a result of the script, the source data were converted into the tabular format of the *Pandas* library. For the convenience of extracting information from the table, a temporary index was formed. After conversion, the result table was ready for further work. It was saved into the open CSV file format.

The next step was to prepare a data sample for modeling. The total sample was n-dimensional arrays, where the last dimension determined the number of data blocks. In neural network models, it is accepted to divide the samples into training, verification, and test ones. Usually 80 % is allocated to the training sample and 20 % to the rest. In this case, it was decided to form samples by time periods: the time period

of 2011–2017 was taken for the training sample, 2018 – for the verification sample and 2019–2020 – for the test sample. Such an approach to sampling excludes data mixing, which would inevitably lead to their imbalance (for example, there may be more data for winter or spring in the training sample than for other seasons), which is not acceptable for the considered task.

The next step was to determine the elements of the sample arrays. For this, the concept of the *data block size* was introduced. Consider *the data block size* as a sequence of measurements that fits into a time interval measured in days. For example, a data block size of 3 days means that it contains 3 * 8 = = 24 measurements.

The input model data are X – wind speed, and output data are Y – meteorological station index. To reach the best neural network training results, it is usually required to normalize a dataset or to convert it into the one hot encoding (OHE) format. An analysis of the entire database showed that wind speed values were in the range 0–19 m/s. Therefore, to represent wind speed in OHE, it is sufficient to use a vector of 20 elements. The meteorological station index is also converted into OHE of 4 elements (by number of stations). For example, for a data block size of 2 days, the following vector sizes are obtained:

– for X: 20 * 2 * 8 = 320 elements;
– for Y: 4 elements.

Modelling

The current work is based on the results from [5]. In [5], the optimal parameters of the neural network model for meteorological time series were selected. The values were as follows:

– batch size = 40;
– value for dropout layers = 0.1;
– activate function = 'relu';
– learning rate = 0.000005;
– epoch count = 50;
– optimizer = 'Adam';
– loss = 'categorical crossentropy';
– metrics = 'accuracy'.

Since the main purpose of the work is to improve the forecast indicators by using one-dimensional convolutional layers, then all the given basic parameters of the neural network remain unchanged.

The optimal network model based on fully connected layers only, taken from [5], is shown in Fig. 1.

Next, we performed experiments with one-dimensional convolutional layers. To do this, the neural network used not only the one-dimensional convolution function *Conv1D*, but also the functions: *SpatialDropout1D* and *MaxPooling1D*. *Keras* also has the *AvaragePooling1D* function, but as experiments showed, it is not suitable for the solving task.

Here you can also introduce the concept of a block for one-dimensional convolution, which consists of a layers' sequence: *SpatialDropout1D*, *Conv1D* and *MaxPooling1D*. Experiments with enumerating parameters in functions showed that the value of the *pool_size* parameter in *MaxPooling1D* should be 2, the value of the *filters* parameter in *Conv1D* should be 10, and the value of *kernel_size* in *Conv1D* should be 5.

This model consists of a sequence of four fully connected blocks. The fully connected block refers to *Dense* and *Dropout* layers' sequence. Further increase in neurons and blocks count does not improve accuracy rates. Note that in the learning process, using the callback, the best result of the arithmetic mean of the accuracy on the verification and test samples is fixed. The best result of the model training is shown in Fig. 2.

Figure 3 shows the neural network model consisting of one convolutional block and two fully connected blocks. Since convolutional layers, unlike fully connected ones, work only with a two-dimensional tensor, it is necessary to use a *Reshape* layer before, and a *Flatten* layer after it.

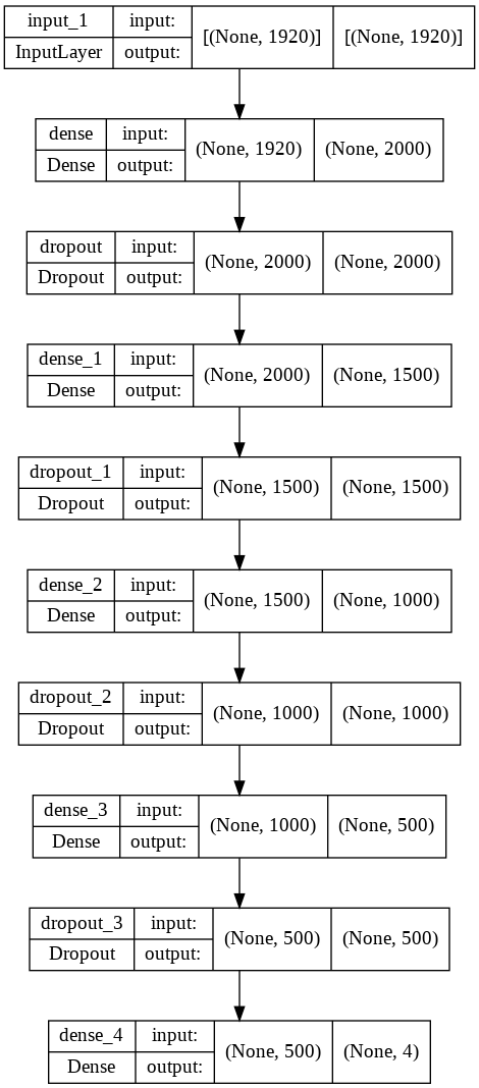

Fig. 1. The model based on fully connected layers only

```
The best accuracy on validation and test sets = 69.38 %

Overall accuracy on the test set: 69.58 %
------------------------------------------------------------
Station classification accuracy: Akhty 65.0 %
Errors on stations:
Derbent - 15
Kochubey - 1
Makhachkala - 5
Station classification accuracy: Derbent 68.33 %
Errors on stations:
Akhty - 13
Kochubey - 0
Makhachkala - 6
Station classification accuracy: Kochubey 85.0 %
Errors on stations:
Akhty - 1
Derbent - 1
Makhachkala - 7
Station classification accuracy: Makhachkala 60.0 %
Errors on stations:
Akhty - 2
Derbent - 10
Kochubey - 12
```

Fig. 2. The best result of the model based on fully connected blocks only training (screenshot from Google Colaboratory notebook)

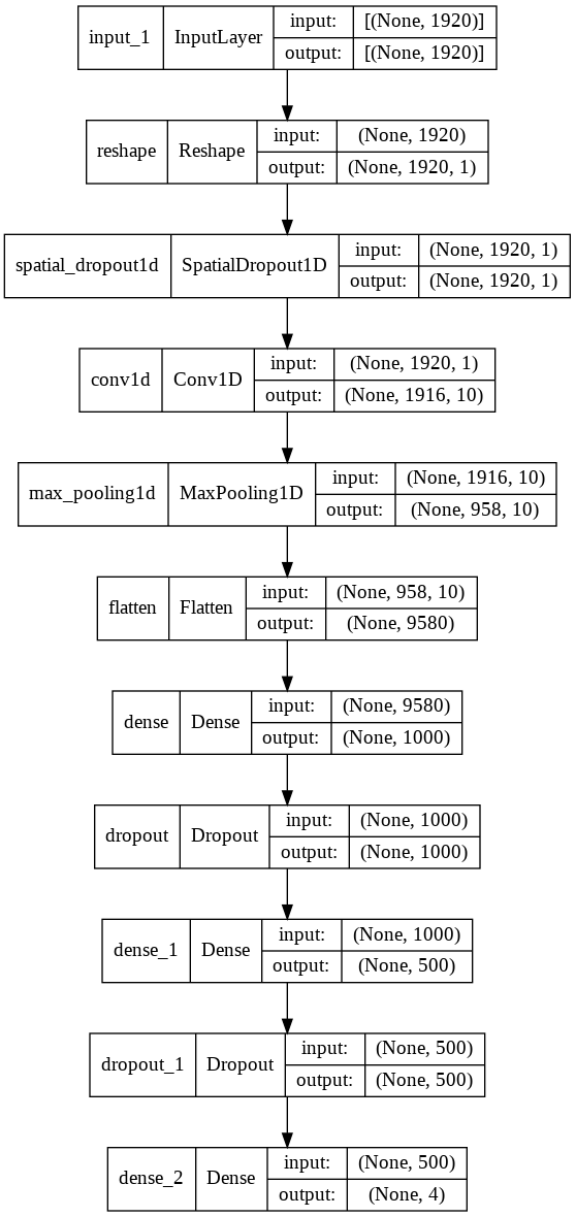

Fig. 3. The model based on one convolutional block and two fully connected blocks

The best result of training the model based on one convolutional block and two fully connected blocks is shown in Fig. 4. The figure shows that the training result only by adding one convolutional block was improved by more than 3 %. Next, you can experiment with the number of fully connected and convolutional blocks, and see what the accuracy limit of the forecast can be achieved.

Experiments with the number of fully connected and convolutional blocks led to the final version of the neural network model, which had the best learning result. The model consists of a sequence of two convolutional blocks and four fully connected ones, and allows you to get result that is another 2 % better than the previous one (Fig. 5).

All the models discussed above have a sequential architecture. The idea arose to try to design and test models with parallel branches and their subsequent merging through the concatenate layer. Obviously, each branch must have a difference in data processing. In parallel model branches, it is logical to vary the parameters of the *Conv1D* (*filters, kernel_size*) and *MaxPolling1D* (*pool_size*) *layers*.

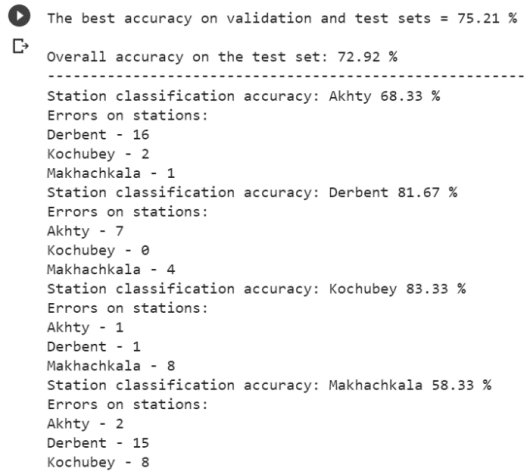

Fig. 4. The best result of the model based on one convolutional block
and two fully connected blocks training (screenshot from Google Colaboratory notebook)

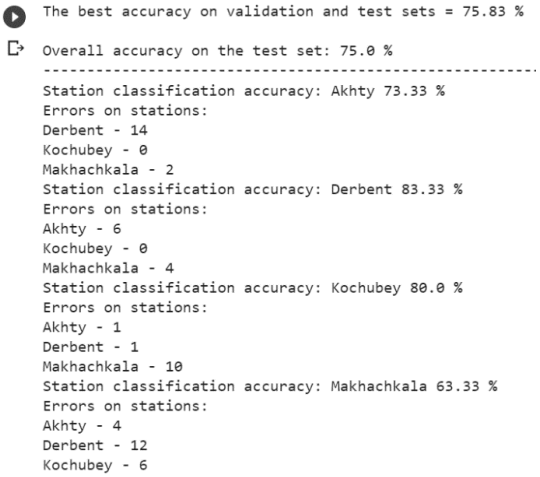

Fig. 5. The best result of the model based on two convolutional blocks
and four fully connected blocks training (screenshot from Google Colaboratory notebook)

As a result of many experiments, the optimal model of the following architecture was adopted. After the initialization and reshaping layer, parallelization into three branches followed, where each branch was a model from Fig. 3 – one convolutional and two fully connected blocks. The branches differed only in the *kernel_size* parameter, which changed with values 3, 5 and 7. After merging the branches in the *Concatenate* layer, a sequence of four fully connected blocks was added to the model, completely in accordance with the model in Fig. 1.

All the completed it is possible to improve the learning result by more than 3 % (Fig. 6).

In addition, the results of classification accuracy by objects turned out to be more balanced than in previous models. This is especially noticeable if we summarize the learning results for the four considered models in a single table (Table 1).

Table 1 shows the effectiveness of using one-dimensional convolutional layers in parallel branches and varying the *kernel_size* parameter in the *Conv1D* layer.

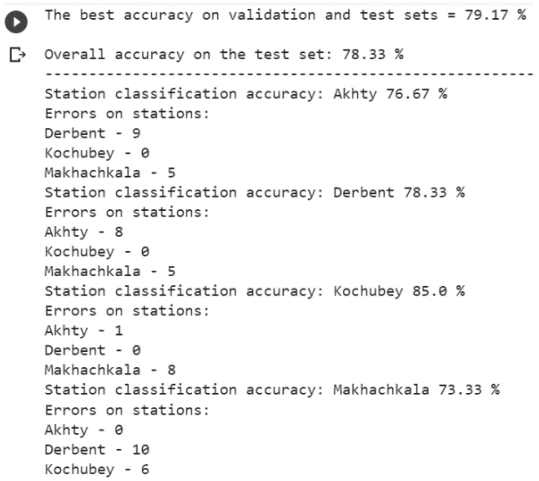

Fig. 6. The best result of the model based on tree parallel brunches training (screenshot from Google Colaboratory notebook)

Table 1

Comparison of the model accuracies, %

	M1	M2	M3	M4
OVERALL	69.58	72.92	75.0	78.33
Akhty	65.0	68.33	73.33	76.67
Derbent	68.33	81.67	83.33	78.33
Kochubey	85.0	83.33	80.0	85.0
Makhachkala	60.0	58.33	63.33	77.33

Conclusion

As can be seen from the results of the presented work (Table 1), when processing the wind speed time series, the use of one-dimensional convolutional networks significantly improves the accuracy of the forecast on the test data set. In addition, a model architecture with parallel branches and variation of the convolution kernel size can be effective and more beneficial in the classification task not only due to the accuracy of the overall forecast, but also thanks to balancing the accuracy of the objects forecast.

As well known, neural network model architectures are directly dependent on the source datasets. It would be incorrect to compare our results with the simulation results for other regions, since the climate is different everywhere, while datasets may differ in data size and completeness.

In the paper, we do not offer a template for solving similar tasks, but a solution to improve the neural network accuracy by adding 1D-convolutional blocks, while also providing some experiments with the possible model architecture variants.

REFERENCES

1. **Catalao J.P.S., Pousinho H.M.I., Mendes V.M.F.** Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable Energy*, 2011, No. 36, Pp. 1245–1251. https://doi.org/10.1016/j.renene.2010.09.016

2. **Paramasivan S.K., Lopez D.** Forecasting of wind speed using feature selection and neural networks. *International Journal of Renewable Energy Research*, 2016, vol. 6 (3), Pp. 833–837. https://doi.org/10.20508/ijrer.v6i3.3855.g6866

3. **Doucoure B., Agbossou K., Cardenas A.** Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *Renewable Energy*, 2016, No. 92, Pp. 202–211. http://dx.doi.org/10.1016/j.renene.2016.02.003

4. **Kobzarenko D.N., Kamilova A.M., Pashtaev B.D.** Issledovanie sezonnyh izmenenij chastotnyh harakteristik skorosti i napravleniya vetra v Primorskom Dagestane [Research the seasonal changes in frequency characteristics of wind speed and direction in Coastal Dagestan]. *YUg Rossii: Ekologiya, Razvitie*, 2020, vol. 15, No. 4, Pp. 152–160. (rus). https://doi.org/10.18470/1992-1098-2020-4-152-160

5. **Kobzarenko D.N.** Analiz vremennyh ryadov – skorostej i napravlenij vetra s pomoshch'yu modelej nejronnyh setej i zadachi klassifikacii [Analysis of time series – wind speeds and wind directions using neural network models and classification task]. *Morskie Intellektual'nye Tekhnologii*, 2021. vol. 1., No. 4 (54), Pp. 127–133. (rus). DOI: 10.37220/MIT.2021.54.4.043

6. **Kiranyaz S., Avci O., Abdeljaber O.** 1D convolutional neural networks and applications: A survey. *arXiv preprint. https://arxiv.org/abs/1905.03554* (*2019*)

7. **Jiao Y., Li N., Mao X., Yao G., Zhao Y., Huang L.** (2021) Pulse recognition of cardiovascular disease patients based on one-dimensional convolutional neural network. *Bio-Inspired Computing: Theories and Applications. BIC-TA 2020. Communications in Computer and Information Science*, Springer, Singapore, 2021, vol. 1363. https://doi.org/10.1007/978-981-16-1354-8_20

8. **Gao C., Wang J., Zhou X., Xiao F., Ma Q.** Classification of lightning electric field waveform based on deep residual one-dimensional convolutional network. *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. ICNC-FSKD 2020. Lecture Notes on Data Engineering and Communications Technologies*, Springer, Cham, 2021, vol. 88. https://doi.org/10.1007/978-3-030-70665-4_179

9. **Xu W., Li H.** Bearing fault diagnosis based on hermitian wavelet and one-dimensional convolutional neural network. *Advances in Intelligent Automation and Soft Computing. IASC 2021. Lecture Notes on Data Engineering and Communications Technologies*, Springer, Cham, 2022, vol. 80. https://doi.org/10.1007/978-3-030-81007-8_41

10. **Yu J., Zhang C., Wang S.** Multichannel one-dimensional convolutional neural network-based feature learning for fault diagnosis of industrial processes. *Neural Comput & Applic*, 2021, No. 33, Pp. 3085–3104. https://doi.org/10.1007/s00521-020-05171-4

11. **Sujanaa J., Palanivel S., Balasubramanian M.** Emotion recognition using support vector machine and one-dimensional convolutional neural network. *Multimed Tools Appl*, 2021, No. 80, Pp. 27171–27185. https://doi.org/10.1007/s11042-021-11041-5

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Кобзаренко Дмитрий Николаевич
Dmitry N. Kobzarenko
E-mail: kobzarenko_dm@mail.ru

Мустафаев Арслан Гасанович
Arslan G. Mustafaev
E-mail: arslan_mustafaev@mail.ru

Гасанова Зарема Ахмедовна
Zarema A. Gasanova
E-mail: cudakharka@yandex.ru

Магомедова Динара Сахратулаевна
Dinara S. Magomedova
E-mail: mdc-101085@mail.ru

Поступила: 30.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023.
Submitted: 30.11.2022; Approved: 26.12.2022; Accepted: 12.01.2023.