

# Use of a convolutional neural network to segment signals of motor operated valves\*

Konstantin I. Kotsoyev<sup>1,3</sup>, Yevgeny L. Trykov<sup>2</sup>, Irina V. Trykova<sup>3</sup>

1 *Bauman Moscow State Technical University, 5 Vtoraya Baumanskaya St., bld.1, 105005 Moscow, Russia*

2 *JSC STC Diaprom, 6 Korolev St., rooms 22-23, 249031 Obninsk, Kaluga Reg., Russia*

3 *KVANT PROGRAMM LTD, 5/12 Zeleny Ave., bld. 3, room B1, 111558 Moscow, Russia*

Corresponding author: Irina V. Trykova ([trykova@kvantprogramm.ru](mailto:trykova@kvantprogramm.ru))

---

**Academic editor:** Yury Korovin ♦ **Received** 24 February 2021 ♦ **Accepted** 9 June 2021 ♦ **Published** 23 September 2021

---

**Citation:** Kotsoyev KI, Trykov YeL, Trykova IV (2021) Use of a convolutional neural network to segment signals of motor operated valves. *Nuclear Energy and Technology* 7(3): 207–213. <https://doi.org/10.3897/nucet.7.73489>

---

## Abstract

Motor operated valves (MOV) are one of the most numerous classes of the nuclear power plant components. An important issue concerned with the MOV diagnostics is the lack of in-process (online) automated control for the MOV technical condition during full power operation of the NPP unit.

In this regard, a vital task is that of the MOV diagnostics based on the signals of the current and voltage consumed during MOV ‘opening’ and ‘closing’ operations. The current and voltage signals represent time series measured at regular intervals. The current (and voltage) signals can be received online and contain all necessary information for the online diagnostics of the MOV status.

Essentially, the approach allows active power signals to be calculated from the current and voltage signals, and characteristics (‘diagnostic signs’) to be extracted from particular portions (segments) of the active power signals using the values of which MOVs can be diagnosed.

The paper deals with the problem of automating the segmentation of active power signals. To accomplish this, an algorithm has been developed based on using a convolutional neural network.

---

## Keywords

Convolutional neural network, time series segmentation, motor operated valves, automated system

---

## Introduction

It often happens in time-series analysis problems that a series is produced by different generation mechanisms. Time series partitioning into internally homogeneous segments is therefore an important issue involved in data mining since it makes it possible to select the key characteristics of a time series from large data arrays in a more compact form (Abonyi et al. 2002).

An example of such time series is signals of the MOV active power during “opening” and “closing” operations, and diagnostics consists in partitioning of signals into segments and further extraction of ‘diagnostic signs’ (numerical values based on which conclusions are made as to the MOV serviceability or unserviceability) from each segment (Matveyev et al. 2009). Each segment is responsible for the actuation of particular MOV parts and components and has specific features of its own.

\* Russian text published: *Izvestiya vuzov. Yadernaya Energetika* (ISSN 0204-3327), 2021, n. 2, pp. 158–168.

Such partitioning is manual in most MOV diagnostics systems, this taking more time and making it impossible to automate diagnostics. To address this issue, a deep neural network has been proposed which segments automatically the active power signal.

## Problem statement

In accordance with methodology (MT 1.2.3.02.999.0085-2010), MOV diagnostics is based on a set of numerical values at representative points and in particular time intervals of the active power signal in the MOV actuation cycle. The representative points in Fig. 1 have been selected with regard for the factors affecting the MOV technical condition and the active power signal changes, and based on the valve gate motion cycles.

The drawback of this approach is the human-dependent algorithm of identifying time intervals. Therefore, automation of the MOV diagnostics process required automatic partitioning of the active power signal into segments so that to extract further from these the numerical values that characterize the MOV technical condition.

It was shown in (Ronneberger et al. 2015) that artificial neural networks coped well with the time series segmentation.

## Initial data

Electrical current and voltage parameters from the MOV motor stator windings (for three phases) were used to calculate the active power signal. The active power was calculated using the formula

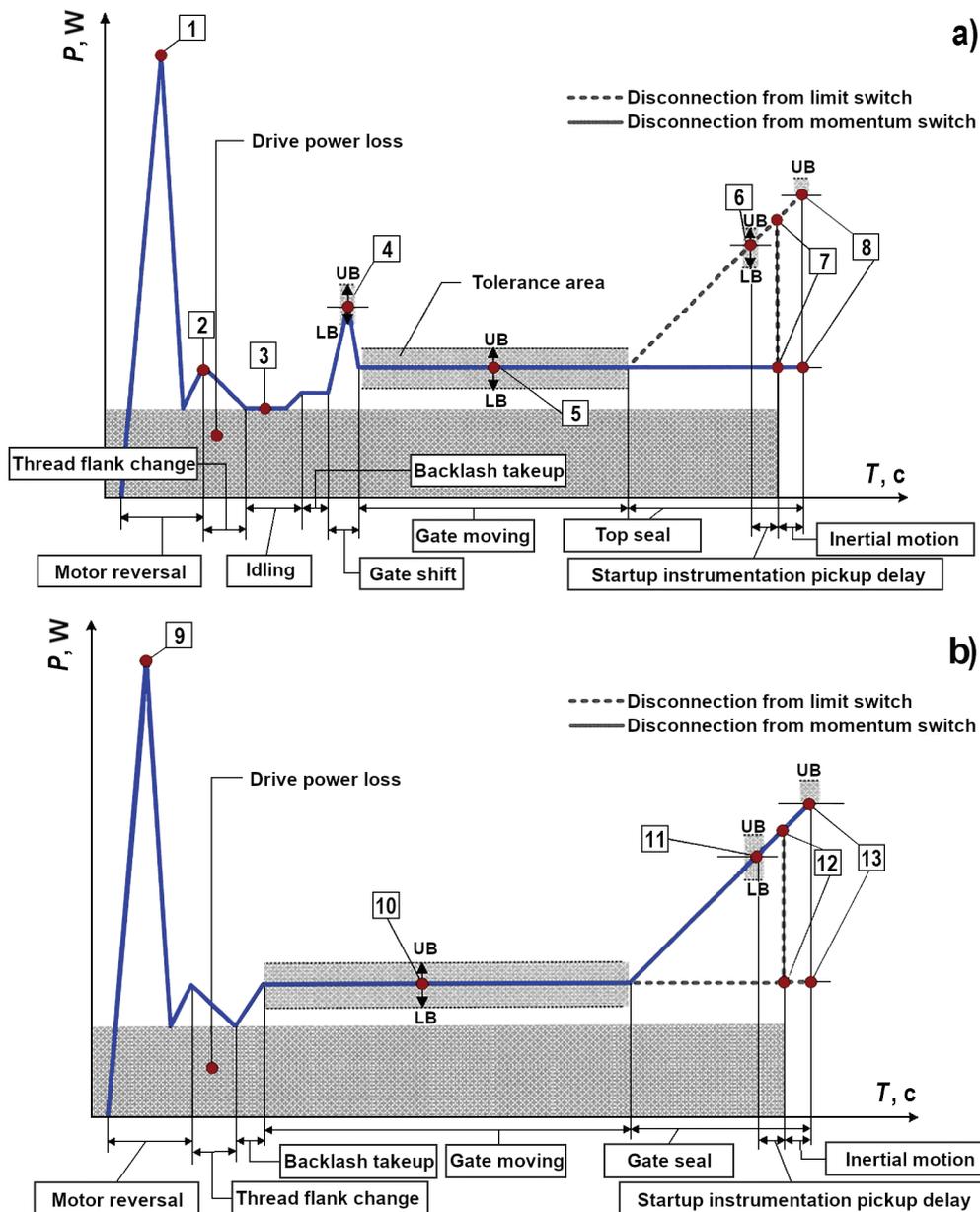


Figure 1. Representative points of an active power signal: a) gate opening, b) gate closing; UB – upper bound; LB – lower bound.

$$P(t) = \frac{1}{T} \int_t^{t+T} u(\tau) \cdot i(\tau) dt, \quad (1)$$

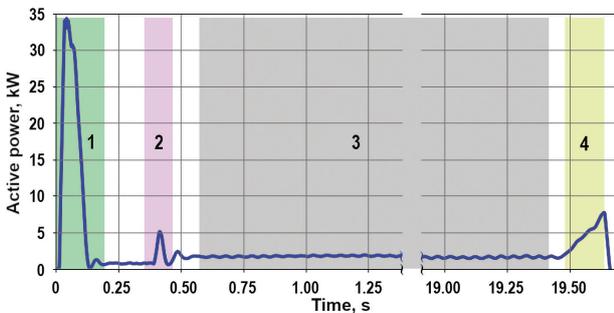
where  $T$  is the carrier frequency cycle (50 Hz); and  $u(\tau)$ ,  $i(\tau)$  are the voltage and current values at time  $\tau$  respectively.

## Determination of segments for the neural network training

The following time intervals were selected for the neural network training:

- motor reversal;
- gate shift (for the ‘opening’ signal);
- gate moving;
- gate seal (for the ‘closing’ signal).

The result of the segmentation is therefore the partitioning of the active power signal into four segments (Fig. 2).



**Figure 2.** Segmentation of an active power signal: 1 – motor reversal; 2 – gate shift (‘opening’); 3 – gate moving; 4 – gate seal (‘closing’).

For the network training, the available active power signals were integrated into a unidimensional array and ‘cut’ into sections of the length 100000. These sections were marked in an interactive mode using a code written in the Python language where the respective class was set to match each point in the signal. Further, a target vector was generated using the One hot coder. The coder takes a column with categorical data and creates several new columns for this. Numbers are replaced by unities and zeroes depending on which value is specific to the given column. In our case, there were five columns which denote

- four classes (see Fig. 2) with the marks 1 through 4 matching the segment number and colored grey;
- one class with the mark ‘No\_label’ (the remaining signal portions of no interest in terms of analysis and uncolored, see Fig. 2).

As a result, the initial data is represented by a set of 571 portions of an equal length and their respective masks with designation of each pixel in the signal belonging to the class (segment) in Fig. 2.

**Table 1.** Segment data

Class	Revers	Podriv	Flat	Stop	No_label
Respective time interval (segment)	Motor reversal	Gate shift	Gate moving	Gate seal	Signal portions not considered in analysis
Pixel ratio, %	0.69	0.68	95.93	0.38	2.32
Number of segments in sample	333	121	330	137	838
Sample size	571				

For clarity, Table 1 presents data on the segments marked and their quantity and ratio.

## Network architecture

The U-Net network proposed in (Ronneberger et al. 2015) and generated in 2015 for segmentation of biomedical images was taken as the basis. Its architecture represents a convolutional network modified such that it could handle less examples (training patterns) and would make segmentation more accurate.

The network consists of an encoder and a decoder connected in series. The encoder is responsible for capturing different features in various scales, and the decoder uses these features to build the final segmentation map. A distinction of this model is that it includes ‘skip-connection’ components which connect the decoder and encoder parts in each scale, that is, the symmetrical encoder output and the output of the preceding decoder layer are connected for the tensor transmission to the decoder input. These layers make it possible to use repeatedly the maps of features from any scale in the decoder, this leading to a more detailed segmentation.

Batch normalization was also added after all convolutional layers which improved the convergence process and the training rate. Besides, it helps monitoring the network weights since their values are at all times saved in the limits of rated values.

## Loss functions

As can be seen in Table 1, there is an unbalance of classes clearly observed in data. Unbalance of data is common to computer-aided training problems such as segmentation and classification. In such situation, where all data is used as is, the classifier will be highly likely to demonstrate a biased capacity in favor of the most represented classes.

Various loss functions were studied to address this issue and to achieve the best segmentation quality.

## Cross entropy

The most commonly used loss function for an image segmentation problem is the cross entropy loss which investigates each pixel individually and compares the class predictions with the given target vector.

Since the cross entropy loss evaluates the class predictions for each pixel vector individually and then undertakes averaging over all pixels, it is essentially states that the training of each pixel in the image is identical. This poses a problem since the classes under consideration have an unbalanced presentation for the sample.

In (Long et al. 2015), it is proposed that this loss is weighed for each output channel so that to oppose the unbalance of classes which the dataset contains. The formula for the categorical cross entropy weighed by classes is written as follows:

$$WCE = -\frac{1}{N} \sum_i \omega_i \sum_j y_{ij} \lg p_{ij}, \quad (2)$$

where  $N$  is the number of classes;  $y$  is the actual value of the class the pixel belongs to; and  $p$  is the predicted value of the class for the pixel.

A loss weighing pattern is discussed in (Ronneberger et al. 2015) for each pixel so that there is a greater weight at the boundaries of the segmented items. This loss weighing pattern has helped the U-Net model segmenting cells in biomedical images in a discontinuous manner such that certain cells can be easily identified in the binary segmentation map.

The final formula for the pixel-weighted categorical cross entropy takes therefore the form

$$PWCE = -\frac{1}{N} \sum_i \sum_j (\omega_i + \omega_{ij}) y_{ij} \lg p_{ij}, \quad (3)$$

$$\omega_i = \lg \frac{n - n_i}{n_i}, \omega_{ij} = \omega_0 \exp \left[ -d_{ij}^2 / (2\sigma^2) \right],$$

where  $\omega_i$  is the weight of the class which was calculated using the formula proposed in (Xiaoya 2020);  $n_i$  is the number of elements in the  $i^{\text{th}}$  class;  $n$  is the total number of elements;  $d$  is the distance to the nearest boundary; and  $\omega_{ij}$  is the pixel weight.

This strategy can be used to control the segmentation results both at the class level and at the pixel level with the loss function adjustment as desired.

Fig. 3 shows how the pixel weights calculated using formula (3) look like.

## Dice coefficient

Dice coefficient described in (Milletari 2016) and presented for the first time in (Sørensen 1948) was also considered as a loss function in (Ronneberger et al. 2015). This function has proved itself to perform well in solving semantic segmentation problems with extremely unbalanced classes. This indicator is in the range  $[0, 1]$  where the Dice coefficient, equal to unity, means an ideal and complete overlap. Dice coefficient was initially developed for binary data and can be calculated using the formula

$$Dice = 2|A \cap B| / (|A| + |B|). \quad (4)$$

Here, the numerator is a doubled set of the elements shared by sets  $A$  and  $B$ , and the denominator is the sum of the quantities of the elements in these sets.

In the event of the Dice coefficient estimation on predicted segmentation masks, one can approximate  $|A \cap B|$  as the element-wise multiplication between the prediction and the target mask, and then sum up the resultant matrix.

Since the target mask is binary for each class, then all pixels from the prediction, which are not “activated” in the target mask, are nulled effectively. For the rest of the pixels, in essence, less valid predictions are penalized; a higher value of this expression leads to a better Dice coefficient.

A simple sum is used by some researchers (Drozdal et al. 2016) to quantitatively estimate  $|A|$  and  $|B|$ , while others (Xiaoya 2020) prefer to use the sum of squares for this computation. Preference was given in this paper to the sum of squares since this loss function offers a better convergence. In this connection, the formula for the Dice coefficient calculation takes the form

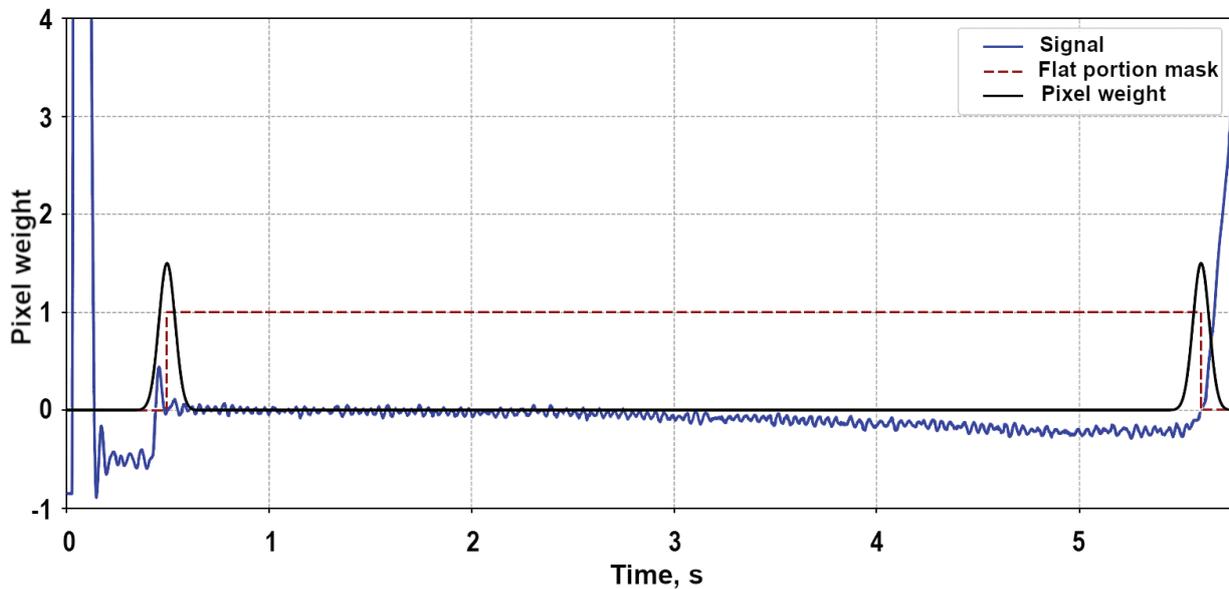


Figure 3. Diagram of pixel weights.

$$Dice = \frac{2 \sum_i y_i \cdot p_i}{\sum_i y_i^2 + \sum_i p_i^2}, \quad (5)$$

where  $y_i$  and  $p_i$  are the actual and predicted probabilities of the class membership.

Then the loss function is determined as

$$Dice\_loss = 1 - Dice. \quad (6)$$

As to the neural network's output data, the numerator in (5) presents the common activations between the prediction and the target mask, as the denominator takes into account the number of the activations in each mask individually. This leads to the normalization of losses in accordance with the target mask size, so no  $Dice\_loss$  hampers the training of classes with a smaller spatial representation in the input data.

Similarly to formulas (3) and (4), we shall introduce the notion of pixel-weighted  $Dice\_loss$ :

$$PWDice\_loss = Dice\_loss + \frac{1}{N} \sum_i \sum_j \omega_{ij} \cdot (y_{ij} - p_{ij})^2. \quad (7)$$

## Evaluation of the neural network quality

A function called Intersection over Union ( $IoU$ ), also known as the Jaccard index and being one of the most

commonly used indicators in semantic segmentation, was used as the metric for the neural network quality evaluation.  $IoU$  is the overlapping region between the predicted segmentation and the actual segmentation divided into the union region between the former and the latter. This indicator is in the range  $[0, 1]$  (0–100%) where zero means no overlapping and unity means the complete overlapping of segmentation. For a multiclass segmentation, the mean value of  $IoU$  ( $Mean\_IoU$ ) is computed by taking  $IoU$  of each class and averaging these.

The formula for the  $IoU$  coefficient determination bears resemblance to Dice coefficient but differs in the denominator and looks as follows:

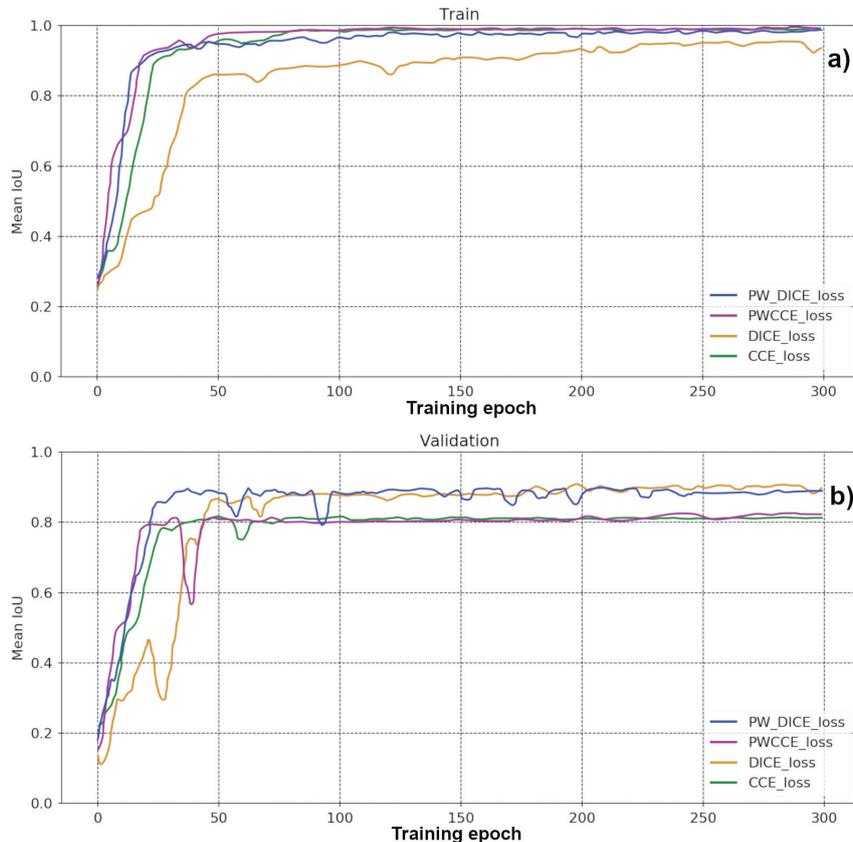
$$IoU = |A \cap B| / |A \cup B|, \quad (8)$$

where  $|A \cap B|$  is the intersection of objects  $A$  and  $B$ ; and  $|A \cup B|$  is their cohesion.

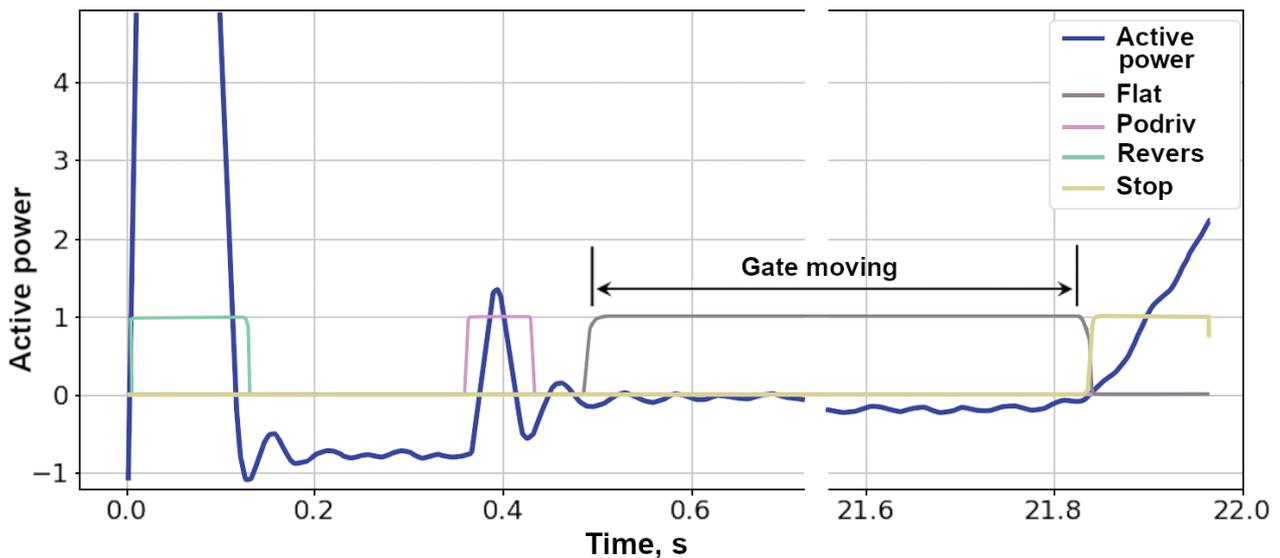
## Training

The neural network was trained with a fixed set of hyperparameters for all tests. The Adam algorithm (Kingma and Ba 2017) with a lot size of 20 and a training rate of 0.001 was selected as the optimizer. Each time, after 10 epochs with no improvement, the training rate was reduced by 20%.

Fig. 4 presents the results of the network training based on training and validation samples using different loss functions.



**Figure 4.** Mean  $IoU$  metric during network training with different loss functions (see formulas 8, 3, 7 and 2): **a)** – training sample; **b)** – validation sample.



**Figure 5.** Results of the active power signal segmentation by a neural network.

A categorical cross entropy leads to better results on the training sample but to worse results during validation, this indicating a better generalizing capability of the *Dice* function.

As can be seen in the diagrams above, adding pixel weighing to the loss function leads to a better convergence of the network and, in the event of the weighed *PWDice-loss*, the values for the training sample of the *Mean\_IoU* metric are 5 to 10% as large as those for *Dice\_loss* without weighing. However, the values of the *Mean\_IoU* metric for the test sample of these functions are approximately equal.

The values of the *IoU* metric for each class (see Table 1) computed by the network with different loss functions are presented in Table 2 for a more detailed evaluation of the segmentation results.

The neural network predicts the probability of a particular class for each time point of the active power signal. Therefore, by selecting a set of the active power signal points, the probability for which is close to unity in the Flat class, one can state that these points belong to the Stroke segment.

An example of the active power signal segmentation is presented in Fig. 5. The lines show the probabilities for the classes (Flat, Podriv, Revers, Stop). By isolating these probabilities with a threshold of, e.g., over 0.95, we get the boundaries of the segments.

## References

- Abonyi J, Szeifert F, Babuska R (2002) Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. In: IEEE Systems, Man and Cybernetics, Part B (Cybernetics), vol. 32, no. 5, 612–621. [Oct. 2002] <https://doi.org/10.1109/TSMCB.2002.1033180>
- Drozdal M, Vorontsov E, Chartrand G, Kadoury S, Pal Ch (2016) The Importance of Skip Connections in Biomedical Image Segmentation. arXiv:1608.04117v2 [cs.CV]. [22 Sep, 9 pp.] In: Carneiro G et al. (Eds) Deep Learning and Data Labeling for Medical Applications. DLMIA 2016, LABELS 2016. Lecture Notes in Computer Science, vol 10008. Springer, Cham, 179–187. [https://doi.org/10.1007/978-3-319-46976-8\\_19](https://doi.org/10.1007/978-3-319-46976-8_19)
- Kingma DP, Ba J (2017) Adam: A Method for Stochastic Optimization. arXiv:1412.6980v9 [cs.LG]. 30 Jan, 15 pp.

**Table 2.** *IoU* by classes (in percent terms) for the test sample

Class	Loss function			
	<i>PWDice_loss</i>	<i>PWCCE_loss</i>	<i>Dice_loss</i>	<i>CCE_loss</i>
Revers	98.12	97.98	97.68	<b>98.28</b>
Podriv	81.32	71.44	<b>92.60</b>	73.21
Flat	99.65	99.64	<b>99.73</b>	99.14
Stop	69.38	62.61	<b>71.65</b>	63.01
No_label	88.49	87.99	<b>90.06</b>	87.32

## Conclusions

A novel approach to segmenting the MOV active power signals has been developed and investigated using a convolutional neural network. It has been found that the *Dice\_loss* loss function allows achieving the best results.

The neural network has shown high-quality results and makes it possible to automate the MOV diagnostics process, to avoid the human factor effects, to increase greatly the diagnostics rate, and to detect the MOV failures, as well as to exclude potential errors caused by the human factor effects.

Thanks to automating the process of partitioning the active power signals into segments, the MOV technical condition diagnostics can be undertaken both offline and online.

- Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. arXiv:1411.4038v2 [cs.CV]. 8 Mar 2015, 10 pp. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 3431–3440. <https://doi.org/10.1109/CVPR.2015.7298965>
- Matveyev AV, Zhidkov SV, Adamenkov AK, Galivets YeYu, Usanov DA (2009) An integrated approach to diagnostics of motor operated valves as applied to lifetime management problems. *Armaturstroyeniye* 2(59): 53–59. [in Russian]
- Milletari F (2016) V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv:1606.04797v1 [cs.CV]. 15 Jun, 11 pp.
- MT 1.2.3.02.999.0085-2010 (2010) Diagnostics of Motor Operated Pipeline Valves. Methodology. Rosenergoatom Publ., Moscow, 239 pp. [in Russian]
- Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597v1 [cs.CV]. 18 May, 8 pp.
- Sørensen Th (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish Cc of the vegetation on Danish commons. *Biologiske Skrifter* 5: 1–34.
- Xiaoya Li (2020) Dice Loss for Data-imbalanced NLP Tasks. arXiv:1911.02855v3 [cs.CL]. 29 Aug, 12 pp.