

# Deep Learning Algorithms for Signal Recognition in Long Perimeter Monitoring Distributed Fiber Optic Sensors

A. V. Makarenko<sup>a, b, 1</sup>

<sup>a</sup> Constructive Cybernetics Research Group  
P.O.Box 560, Moscow, 101000 Russia

<sup>b</sup> Institute of Control Sciences, Russian Academy of Sciences  
ul. Profsoyuznaya 65, Moscow, 117977 Russia

Received April 27, 2016; in final form, July 28, 2016.

**Abstract.** In this paper, we show an approach to build deep learning algorithms for recognizing signals in distributed fiber optic monitoring and security systems for long perimeters. Synthesizing such detection algorithms poses a non-trivial research and development challenge, because these systems face stringent error (type I and II) requirements and operate in difficult signal-jamming environments, with intensive signal-like jamming and a variety of changing possible signal portraits of possible recognized events. To address these issues, we have developed a two-level event detection architecture, where the primary classifier is based on an ensemble of deep convolutional networks, can recognize 7 classes of signals and receives time-space data frames as input. Using real-life data, we have shown that the applied methods result in efficient and robust multiclass detection algorithms that have a high degree of adaptability.

**Keywords:** deep learning, fiber optic vibration sensors, signal recognition, events classification, t-SNE visualization.

## 1. Introduction

Today, distributed fiber optic vibration sensors are used in many areas and undergo rapid development, improving their sensitivity and selectivity characteristics [1]. Due to a number of design advantages (feasibility, full passivity, negligible response to electromagnetic radiation, compatibility with off-the-shelf telecommunication cables, etc.), fiber optic sensors are very common in monitoring and security of long perimeters, one of the key applications being monitoring of major pipelines.

One of the types of fiber optic systems operates based on time-domain detection of backward Rayleigh scattered light of short laser pulse injected into the cable [1]. Coherence reflectometer uses a narrowband high-stable reference laser as the light source (see Fig. 1 for the schematic structural and functional diagram). This allows to sum the intensities of the backscattered signals with phase delay factored in. Meanwhile, vibrations in the

---

<sup>1</sup>E-mail: avm.science@mail.ru

environment, where the sensor operates, cause micro-vibrations in the optic fiber of the sensor (shift amplitudes from 50 nm), displacing Rayleigh scattering centers relative to each other. As a result, spatially localized disturbances of the sensor cause local modulation of coherence reflectogram intensity.

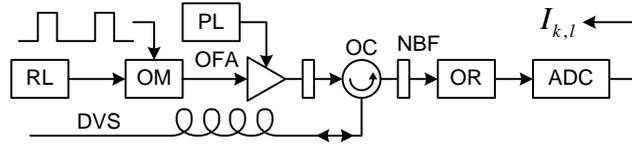


Figure 1. Schematic structural and functional diagram of the coherence optical time domain reflectometer: RL – reference laser; OM – electrooptical light modulator; OFA – optical fiber amplifier; PL – pump laser; NBF – narrow-band filter; OC – optical circulator; DVS – distributed optical fibre vibration sensor; OR – optical receiver; ADC – analog-to-digital converter;  $I_{k,l}$  – intensity of the signal registered from the  $l$ th spatial element of the sensor at moment  $k$ .

Analyzing time-frequency and spatial features of the intensity of local modulation in the coherence reflectogram allows to build algorithms for detection and classification of vibration and acoustic fields disturbances in the vicinity of monitored perimeters [2]. Note that there are complicating factors in building such algorithms. The major ones are:

- deviations in local temperature of the sensor causes ghost modulations of the reflectogram;
- deviations in laser light phase (frequency) cause local sections of the sensor to become temporarily insensitive;
- great variety of changing possible signal portraits of detected events requires ingenious approaches to building dictionaries for identification of events;
- broad range of changing mechanical and acoustic properties of the environment where the sensor operates leads to significant changes in the signal portraits of recognized events.

The abovementioned factors as well as intensive signal-like jamming make for a complex signal-jamming environment that the event recognition algorithms have to process. Another complicating factor is the stringent requirements to monitoring and security systems at critical facilities in terms of type I and II error rates in detection and classification of events in the monitored area.

Therefore, a considerable depth of adaptability is essential to efficient and reliable operation of the recognition algorithms. Meeting these requirements is quite difficult if standard methods of algorithm synthesis are used (e.g., see [3]). Moreover, our research has indicated that a number of machine learning algorithms [2, 4] are inadequate for this task as well. Hence, to solve this problem, we have built a distributed fiber optic sensor system using deep learning algorithms based on an ensemble of convolutional neural networks.

## 2. The Problem Description

We synthesize a signals recognition algorithm for a distributed fiber optic monitoring and security system for major pipelines (developers PetroLight Company and OMEGA Company, Moscow, Russia). Key specifications: monitoring perimeter length (one optical sensor) is 50 km; space step of the sensor is 2 m; ADC resolution is 16 bit; signal sampling rate for one spatial channel of the sensor is 1666 Hz.

We have defined the following classes of detected and classified signals (events) according to the specific purpose of the system:

- 0 – "natural and technogenic jamming";
- 1 – "liquid spilling into the ground from pipeline leakage";
- 2 – "digging with hand tools";
- 3 – "digging with heavy excavation equipment";
- 4 – "drilling into the pipeline wall";
- 5 – "welding onto the pipeline wall";
- 6 – "grinding the pipeline wall".

Note that the signal (event) classes defined above address the whole range of control tasks in the monitored area of major pipelines and allow to detect conventional forms of illegal activity.

The Fig. 2 shows power spectrums of the investigated signals averaged in time (non-coherent combining interval  $\approx 20$  min) and normalized in 0 to 10 dB range.

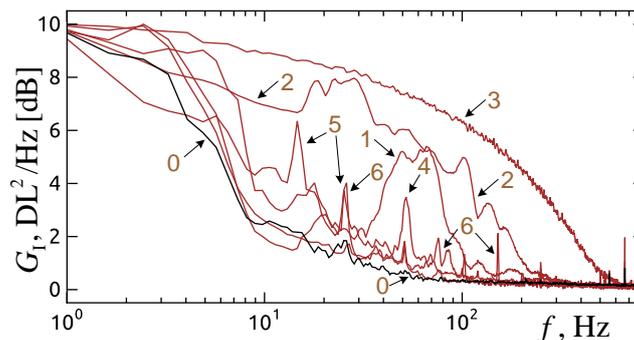


Figure 2. Power spectrums of the investigated signals averaged in time (computed using the Hamming window), DL is digital level of ADC, digits 0, ..., 6 represent classes of the investigated signals.

The spectrums in Fig. 2 are typical of the detected signals and, as can be clearly seen, considerably different from each other. In light of this, we have attempted to synthesize a classifier based on the shape of the integral spectrum using Support vector Machine, Random Forest and Fully Connected Neural Networks. Informative features were synthesized using "hand-engineered" feature extraction technique. The results have turned out below satisfactory, because the resulting classifiers lack robustness in the changing signal-noise environment. Additional investigations have indicated that, due to the factors presented in the Introduction, the shape of spectrums stabilizes (manifests) only when the generation intervals are longer. At the same time, their shape can significantly change depending on the disturbance source, leading to signal profile variation effect). In fact, the problem described above is quite common when "hand-engineered" feature extraction technique is applied [5].

With regard to these new findings, we have decided to synthesize a detection algorithm based on convolutional neural networks, which have been proven effective for similar applications, e.g., see [6–8].

To train and test the algorithms, we have built a reference dataset of real-life signals and jamming under the conditions similar to normal modes of operation of the instrument. Dataset samples have been recorded in three geographical zones, differing in ground properties, cable installation, major pipeline specifications, and natural and manmade jamming

patterns. The recordings cover several calendar seasons in all three zones, with ground freezing, thawing and watering factored in. It is also worth noting that we have introduced significant variation into the key parameters of signal sources in order to obtain a wider range of signal portraits of recognized events and to enhance the generalization ability of the classifier. For example, to record class 1 signals (Leakage) we have used pipes of various diameters, various liquid pressures and defects of various sizes; for class 5 signals (Welding) we have used electrodes of various diameters and various welding current settings.

### 3. The Target Data Processing Circuit

#### 3.1. The Circuit Structure

The Fig. 3 shows the schematic structural and functional diagram of the synthesized target data processing circuit for primary signal classification.

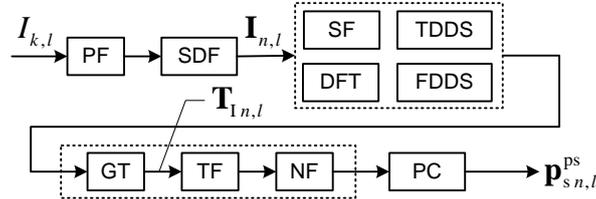


Figure 3. Schematic structural and functional diagram of the target data processing circuit: PF – primary filter; SDF – shaper of data frame; SF – secondary adaptation filters; DFT – digital Fourier transform; TDDS – time domain decision statistics; FDDS – frequency domain decision statistics; GT – generation of primary features array; TF – transformation of primary features; NF – normalization of primary features; PC – primary classifier.

The ADC provides digital sample stream as raw data input to be filtered channel by channel by the primary filter in order to remove components that fall outside the valid signals range. Then, we feed the filtered data into a block that uses the digital sample stream to build single-channel time-space data frames:

$$\begin{aligned} \mathbf{I}_{n,l} &= \{I_{k',l}\}_{k'=k_b}^{k_e} = [I_{k_b,l}, I_{k_b+1,l}, \dots, I_{k_e,l}], \\ k_b &= n K' / f_d, \quad k_e = k_b + K' - 1, \end{aligned} \quad (3.1)$$

where  $n$  and  $K'$  are the number and the size of the data frame,  $f_d = 1, \dots, 8$  is the overlap factor. For each data frame, we compute a decision statistics set for both the time domain (coefficients of kurtosis, skewness, etc.) and the frequency domain (filter bank, etc.). Note that to improve the robustness and efficiency of the recognition algorithms we have introduced a secondary filters, which ensures the adaptability of the algorithms.

Next, we use the decision statistics set to generate a primary features array  $\mathbf{T}_{I_{n,l}}$ , which has a specific structure coordinated with the primary classifier input. We transform this array applying specific rules, normalize and feed it into the primary classifier that provides an estimation of confidence (posterior probability) attributing time-space data frames to one of the signal classes  $\mathbf{p}_{s,n,l}^{ps}$ .

At the stage of defining and analyzing primary features, we have applied nonlinear decrease dimensionality reduction t-SNE method [9]. We have employed a portion of the test set, 35K samples used in total, with the recognized signal classes equally distributed within the set (see Sec. 3.4). We have run PCA analysis and used the first 64 principal components for the source dataset. We have used 3D t-SNE visualization. Fig. 4 shows median cluster

centers, which have been interconnected through minimum spanning tree algorithm, with the distances between graph node minimized.

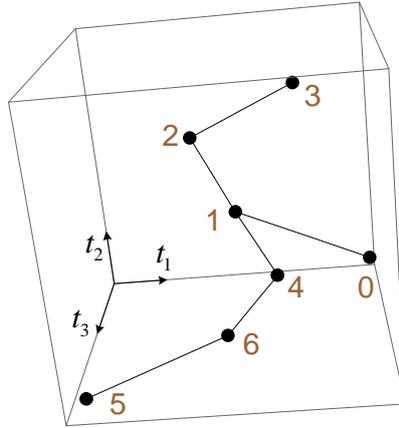


Figure 4. Median t-SNE cluster centers for the signal classes.

Note that the obtained t-SNE tree of cluster centers has proved to be robust against variations in the key parameters of the nonlinear decrease dimensionality reduction algorithm. In some cases, the 0-1 connection changed for 0-4, whereas the rest of the graph remained the same. With regard to the t-SNE algorithm properties, this stability of partition indicates that the data structure in the domain of the selected informative features possesses a high level of invariance.

The Fig. 4 indicates that the 7 signal classes, that we have defined among the identified primary features (specifically, their main entries), can be clearly differentiated between. We see that the signals of classes 1 (Leakage) and 4 (Drilling) look closest to class 0 signals (Background), whereas the signals of classes 3 (Excavation) and 5 (Welding) are the most distinct from the Background, see Table 1.

Table 1. Distances between cluster centers.

#	1	2	3	4	5	6
0	0.486	0.782	1.000	0.551	0.881	0.626
1		0.310	0.575	0.201	0.534	0.312
2			0.307	0.353	0.558	0.440
3				0.545	0.715	0.626
4					0.416	0.153
5						0.281

The results essentially correspond to the shape of the integral power spectrum of signals (see Fig. 2). Later, the analysis of the error matrix of the trained classifier have corroborated the results as well (see Table 2 for example).

### 3.2. The Primary Classifier

The Fig. 5 shows the scheme of the primary classifier structure.

The Fig. 5 shows that the primary classifier is an ensemble of three forward-propagation deep neural networks (C1–C3), where convolution and fully connected layers are consecutively combined. The key difference between the networks is their convolution

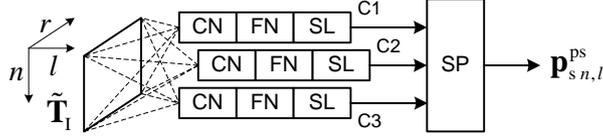


Figure 5. Schematic structural and functional diagram of the primary classifier: CN – convolution network; FN – fully connected network; SL – softmax layer; SP – aggregating rule:  $r = 1, \dots, R_M$ , where  $R_M$  is primary feature vector dimensionality.

kernel configuration. Note that the use of three networks is a compromise between the quality of classifier decisions and its speed of real-time operation. We feed a 2D data blob with the input format of  $N_M \times R_M$  into each of the networks so that the primary classifier processes the spatial input channel by channel. The output layer of each network has 7 neurons (by the number of recognized signal classes) and uses SoftMax activation function:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=0}^6 e^{z_j}}, \quad i = 0, \dots, 6. \quad (3.2)$$

This allows to interpret the values of the output vector  $\sigma(\mathbf{z})$  as event probabilities that form a full group when taken together.

We can obtain partial solutions for C1–C3 networks using a threshold rule:

$$c_{s n, l}^{ps(j)} = \begin{cases} i^* & \sigma(\mathbf{z})_{i^*}^{(j)} \geq (\alpha_c^{(j)})_{i^*}, \\ 0 & \text{otherwise.} \end{cases}, \quad (3.3)$$

$$i^* = \arg \max_i \sigma(\mathbf{z})_{n, l}^{(j)}, \quad i = 0, \dots, 6, \quad j = 1, \dots, 3,$$

where  $\alpha_c^{(j)}$  is the vector set for decision thresholds in  $j$ th network. The most popular and, in a way, the simplest rule for generating a single decision based on partial solutions for C1–C3 networks is two-out-of-three selection. With the transitivity property factored in, it can be defined as follows:

$$c_{s n, l}^{ps} = \max \left\{ c_{s n, l}^{ps(1)} d_{12}, c_{s n, l}^{ps(1)} d_{13}, c_{s n, l}^{ps(2)} d_{23} \right\}, \quad (3.4)$$

$$d_{ij} = \delta \left[ c_{s n, l}^{ps(i)}, c_{s n, l}^{ps(j)} \right], \quad i, j = 1, \dots, 3,$$

where  $\delta[\cdot, \cdot]$  is the Kronecker delta.

The standard approach based on rule (3.4) has a significant drawback, as the information on posterior probability of time-space data frame attribution is lost too early. This puts a critical limitation on the system, because the receptive fields of the primary classifier are separated (isolated) into spatial channels (where the input is a data blob of  $N_M \times R_M$  format), meaning that the classifier cannot either select a signal source based on its space width or differentiate between a moving and a static signal source. Such a simplification is necessary due to real-time classification and hardware limitations that are usually bypassed via secondary data processing, see Sec. 3.3. Note that we are researching algorithms to synthesize classifier for the input format of  $N_M \times L_M \times R_M$ , where  $L_M > 1$ , but the requirements to GPU performance are lower.

In light of the above, we can propose two simple rules for generating a single decision based on partial solutions for C1–C3 networks, where the information on posterior probability of time-space data frame attribution is retained. First, averaging by  $L_2$  norm:

$$\mathbf{p}_{s n, l}^{ps} = \frac{\mathbf{s}}{|\mathbf{s}|}, \quad \mathbf{s} = \sum_{j=1}^3 \sigma(\mathbf{z})_{n, l}^{(j)}, \quad (3.5)$$

second, selecting decision with the highest confidence (in fact  $-L_\infty$  norm):

$$\mathbf{p}_{s_{n,l}}^{\text{ps}} = \sigma(\mathbf{z})_{n,l}^{(g)}, \quad g = \arg \max_j \sigma(\mathbf{z})_{n,l}^{(j)}, \quad j = 1, \dots, 3. \quad (3.6)$$

It is worth noting that we have investigated more complex rules for generating a single decision based on partial solutions for C1–C3 networks as well.

### 3.3. The Secondary Data Processing

The primary classifier output is inadequate to be directly used for generating reports on the situation in the monitored area. The reason for that is twofold. First, as has been explained above, the primary classifier cannot select signal sources based in their space width and cannot identify a moving source. Second, in order to minimize the number of undetected events and to stabilize the false alarm rate, the decisions of the primary classifier need to be glued (as well as filtered) into signal-event tracks that are used to form final event reports. We resolve these issues by using a secondary classifier.

Detailed description of the secondary data processing algorithms and synthesizing them is beyond the scope of this paper, besides we are actively researching this area at the moment. Therefore, we confine ourselves to highlighting several key issues that we have already resolved. We use vectors  $\mathbf{p}_{s_{n,l}}^{\text{ps}}$  as the main input for the secondary classifier, thereby achieving stacking of the primary and secondary classifiers. For the secondary classifier input, we use main-data blob of  $N_H \times L_H \times M_H$  format, where  $M_H$  is determined by the number of recognized signal classes, and we set  $L_H > 1$  to introduce spatial features into the algorithm (see Fig. 6).

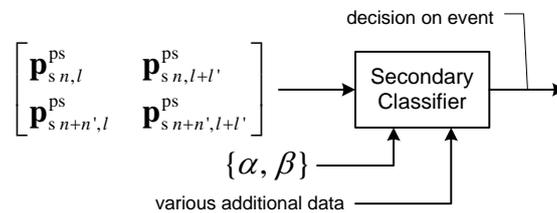


Figure 6. Schematic structural and functional diagram of the secondary classifier:  $\{\alpha, \beta\}$  are the allowed error rates for type I and II errors.

We build the architecture of this classifier based on deep neural networks as well. The decisions of this classifier can be controlled by the operator, who can set the allowed error rates for type I and II errors. Moreover, this classifier operates with various additional data, namely: a priori probabilities of certain event classes being detected at a given time in a specific area; primary classifier output provided by a second optical sensor (back-to-back monitoring of the perimeter by two sensors); decisions provided by other monitoring systems based on other operation principles.

### 3.4. Classifier Training

Prior to defining and analyzing primary features and to training and testing the classifier, we have had an operator mark up the raw dataset of signals, jammings and noises (see Sec. 2) using automation technology. The mark-up consists of defining time-space boundaries of signals, assigning event classes and introducing additional machine-readable metadata. Next, we have separated the data into training and test sets, which do not overlap. The results of the present paper have been obtained on 487K samples in the training set

and 70K samples in the test set. The shares of the event classes in the test set are uniform balanced.

We trained the three deep neural networks C1–C3 separately. We applied the single-label approach (one-hot vector) and the loss function for categorical crossentropy:

$$L^{(j)} = -\frac{1}{K} \sum_{k=1}^K \sum_{i=0}^6 (\mathbf{t}_k)_i \ln \sigma(\mathbf{z})_{ik}^{(j)}, \quad j = 1, \dots, 3, \quad (3.7)$$

where  $K$  is the size of the training set,  $\mathbf{t}_k$  is the label vector tied to the  $k$ th time-space data frame included into the training set. Note that, in principle, at the stage of training, it is possible to introduce an additional component into the objective function to estimate real (financial) losses and gains of certain classifier decisions.

To improve the overall quality, we optimized the architecture of C1–C3 networks (the order, number and configuration of the layers, the number and configuration of the convolution kernels, etc.) applying the differential evolution method [10]. To prevent overfitting, we used *dropout* and  $L_2$  regularization methods [8].

After the primary training and optimization of the classifier, the adjustment of the training and test sets mark-up was carried out by the classifier itself. At this stage, we used rules (3.5) and (3.3) to correct the vector for  $\mathbf{t}$  labels. Next, we performed the final tuning of the primary classifier. Note that we present the results of the t-SNE analysis (see Sect. 3.1) and the classifier quality parameters (see the section below) at the stage of primary training, i.e., before the data set filtering and tuning of weights of the neural networks. We trained the networks for  $\approx 200$  epochs using the SGD method [8].

### 3.5. The Results

To estimate the integral quality of the classifier operation on the test set, we have used standard Accuracy metric:

$$A = \frac{1}{K} \sum_{k=1}^K \begin{cases} 1 & \arg \max_i \mathbf{t}_k = \arg \max_i \mathbf{p}_{sk}^{\text{ps}}, \\ 0 & \text{otherwise.} \end{cases} \quad i = 0, \dots, 6 \quad (3.8)$$

The C1–C3 networks have achieved the following accuracies:  $A^{(1)} \approx 91.20\%$ ,  $A^{(2)} \approx 93.37\%$ , and  $A^{(3)} \approx 91.12\%$ .

We have performed detailed investigation of the primary classifier accuracy and of its components based on the confusion matrix analysis and the density of classifier decisions distribution under various signal-jamming environment conditions. By way of example, the error matrix of the C1 network is given in Table 2.

By way of example, Fig. 7 shows spatial-temporal representation of the C1 network decision after application of rule (3.3) for validation sample #269 (not included into the training set), which contains a class 6 signal.

The Fig. 7 indicates that the C1 network has clearly and accurately identified the time-space frames that contain test signal of class 6, thus enabling the secondary classifier to generate an signal-event track. The parameters of the output track have turned out highly accurate with regard to the reference parameters, leading to a true-positive decision on detection of a class 6 event.

The level of error in the C1 network disproportionately leaning toward class 4 signals correlates with the data from t-SNE analysis (see Fig. 4) and can be explained by the signal

Table 2. Classifier C1: confusion matrix, test set.

Event class		Prediction, %						
		0	1	2	3	4	5	6
Reference, 100%	0	91.80	3.96	0.64	0.14	2.34	0.42	0.70
	1	13.78	79.24	5.38	0.02	1.14	0.28	0.16
	2	4.24	3.34	91.30	0.14	0.66	0.00	0.32
	3	2.36	0.10	0.28	97.08	0.12	0.00	0.06
	4	8.68	0.38	0.22	0.00	89.80	0.28	0.64
	5	3.34	0.14	0.02	0.00	0.30	94.40	1.80
	6	3.42	0.12	0.14	0.00	0.70	0.82	94.80
Prec.		71.93	90.79	93.18	99.69	94.47	98.13	96.26
F1 sc.		80.66	84.62	92.23	98.37	92.07	96.23	95.53

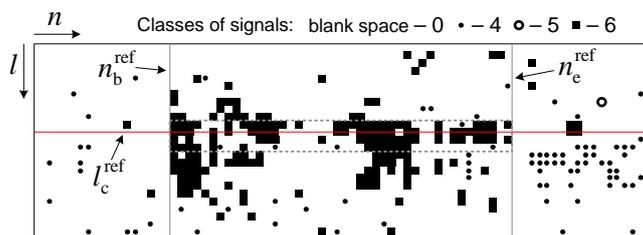


Figure 7. The decision of the C1 network for validation sample #269 (class 6 signal). The mark-up denotes:  $n_b^{\text{ref}}$  and  $n_e^{\text{ref}}$  are reference marks for signal beginning and end;  $l_c^{\text{ref}}$  is reference (central) spatial channel; gray dotted rectangle is the signal-event track generated by the secondary classifier.

portrait of this class being challenging to recognize (see Fig. 2). However, the ensemble of networks C1-C3 suppresses these deviations quite effectively. Note that the probabilities of error in the primary classifier output cannot by any means be considered as final in terms of event decisions, as the system is equipped with the secondary classifier (see Sect. 3.3) that substantially decreases the levels of type I and II errors (as well as the ratio between them), when the decisions on frames are followed by decisions on signal-event tracks and the final event decision.

Finally, it is worth noting that, depending on the network, forward pass through C1–C3 networks takes from 12 to 25 ms for 1000 distance channels computed on GPU NVIDIA GeForce GTX Titan X. We see that the classifier facilitates real-time decisions for a monitored area of 50 km length, with spare hardware capacity that provides opportunity for further development of the algorithms.

## 4. Conclusion

In this paper, we have shown that efficient signal recognition algorithms for distributed fiber optic monitoring and security systems for long perimeters can be created based on deep learning methods. The integral averaged quality of a trained neural network of the primary multiclass detector is no less than 91% of correctly recognized time-space data frames in the detector. We have obtained this estimate without applying the network ensemble or network weights tuning in conditions simulating actual operation conditions of the system to the maximum extent. Unlike other approaches, as for example [3] or when only "hand-engineered" feature extraction technique is used, deep learning methods require less development effort and, most importantly, they allow for highly flexible reorganization of the system in case

of changes in the classification of recognized events and/or in signal-jamming environment parameters.

It is worth noting that we are continuing to improve the parameters of the synthesized algorithms in different ways, among other things, by applying new deep architectures of neural networks.

Moreover, we have shown that nonlinear dimension reduction method t-SNE [9] can be an efficient tool for preliminary analysis of informative features in terms of differentiation of signal classes. This allows to bypass the training of deep neural networks at the preliminary stage of synthesizing primary features, which saves time and hardware capacity.

In conclusion, it is worth noting that the demonstrated approach to developing signal recognition algorithms based on deep learning can be successfully extended to other applications, including radiolocation, hydroacoustics, supersonic or magnetic scanning of materials, etc. This conclusion is supported by the fact that the studied device (coherence optical time domain reflectometer) has nonlinear and nonstationary mode of operation [1] and is used in difficult signal-jamming environments, whereas the algorithms based on convolutional neural networks counteract these negative factors quite successfully.

## 5. Acknowledgments

The authors thank the anonymous referees for their useful comments.

## References

1. A. Masoudi and T.P. Newson, “Contributed review: Distributed optical fibre dynamic strain sensing,” *Review of Scientific Instruments*, vol. 87, no. 1, 2016.
2. S.S. Mahmoud, Y. Visagathilagar, and J. Katsifolis, “Real-time distributed fiber optic sensor for security systems: Performance, event classification and nuisance mitigation,” *Photonic Sensors*, vol. 2, no. 3, pp. 225–236, 2012.
3. Song I., J. Bae, and S.Y. Kim, *Advanced Theory of Signal Detection.*, Springer, 2002.
4. A. K. Fedorov, M. N. Anufriev, A. A. Zhirnov, K. V. Stepanov, E. T. Nesterov, D. E. Namiot, V. E. Karasik, and A. B. Pnev, “Note: Gaussian mixture model for event recognition in optical time-domain reflectometry based sensing systems,” *Review of Scientific Instruments*, vol. 87, no. 3, 2016.
5. L.E. Besaw and P.J. Stimac, “Deep learning algorithms for detecting explosive hazards in ground penetrating radar data,” 2014, vol. 9072, pp. 90720Y–90720Y–11.
6. O. Abdel-Hamid, A.R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, 2014.
7. S. Chen and H. Wang, “Sar target recognition based on deep learning,” in *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, 2014, pp. 541–547.
8. I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” Book in preparation for MIT Press, 2016.

9. L.J.P. van der Maaten and G.E. Hinton, “Visualizing high-dimensional data using t-sne,” *JMLR*, vol. 9, pp. 2579–2605, 2008.
10. K. Price, R.M. Storn, and J.A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization.*, Springer, 2005.

**Andrey V. Makarenko** – was born in 1977, since 2002 – Ph. D. of Cybernetics. Founder and leader of the Research & Development group “Constructive Cybernetics”. Author and coauthor of more than 60 scientific articles and reports. Member IEEE (IEEE Signal Processing Society Membership; IEEE Computational Intelligence Society Membership). Research interests: Analysis of the structure dynamic processes, predictability; Detection, classification and diagnosis is not fully observed objects (patterns); Synchronization and self-organization in nonlinear and chaotic systems; System analysis and math. modeling of economic, financial, social and bio-physical systems and processes; Convergence of Data Science, Nonlinear Dynamics, and Network-Centric.