

Neural Network Classifier of Oil Pollution on the Water Surface when Processing Radar Images

Tatiana Tatarnikova^[0000-0002-6419-0072] and Elena Chernetsova^[0000-0001-5805-3111]

¹ Russian State Hydrometeorological University, 79, Voronezhskaya st., 192007 St. Petersburg, Russia
tm-tatarn@yandex.ru
chernetsova@list.ru

Abstract. The paper proposes a solution to the problem of detecting oil pollution on a monochrome radar image. The detection of oil pollution in the image includes the solution of three tasks: detecting a dark object on image, highlighting the main characteristics of a dark object, classifying a dark object as oil pollution or natural slick. Various characteristics of a dark object are proposed based on the contrast between the object and the background. It is proposed to use a neural network as a classifier. The input parameters of the neural network classifier of the dark image object are proposed. A technique for determining the structure of a neural classifier is presented. An algorithm for testing the selected structure of the neural network for the suitability of classifying the dark area on the image of the water surface as oil pollution or wind slick is proposed. The results of the work of the neural network classifier program for detecting abnormal objects in radar images are demonstrated.

Keywords: Radar Image, Object Detection on the Image, Slick on the Image, Neural Network, Classifier, Oil Pollution, Wind Slick, Water Surface

1 Introduction

The paper proposes a neural network (NN) classifier, which is used in the processing of radar images (RI) of the sea surface. The classifier identifies a dark object on the radar and detects it as an oil slick or wind slick [1].

The presence of an oil film on the surface of the sea reduces small waves due to the increasing viscosity of the upper layer and significantly reduces the energy of backscattering of the signal, therefore, dark regions appear on the radar images [2]. However, blackouts in the image can also occur due to local low-speed winds above the sea surface or the presence of natural sea slicks. Examples of radar images with dark objects are shown in Fig. 1 a, b.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

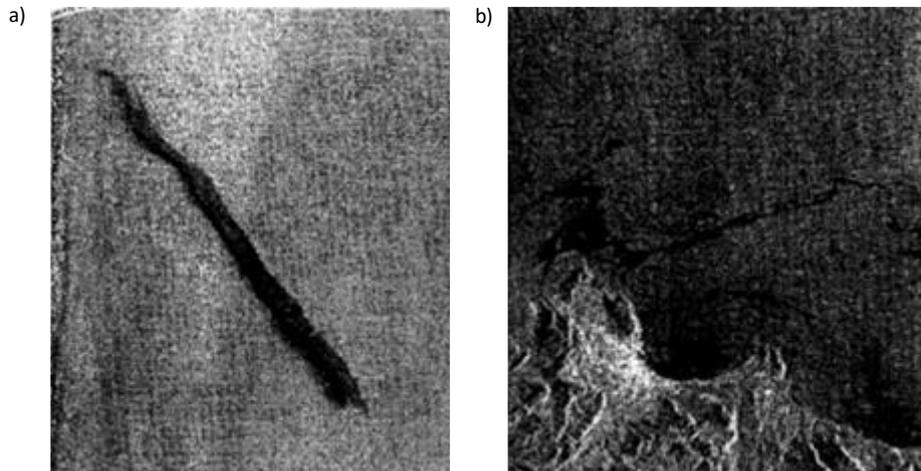


Fig. 1. a) Image of an oil slick b) Image of a natural slick

To classify these types of objects in the image, we created a data bank of parameters for training the neural network. The research used data calculated for 139 objects, of which 71 images with oil spills and 68 images with wind slicks.

2 Selection of Input Parameters for the Designed Neural Network

The contrast between oil pollution and the surrounding background depends on the height of the waves, the amount of spilled oil, wind speed and other factors.

The detection of oil pollution in the image includes the solution of three tasks:

- Dark object detection
- Highlighting the main characteristics of a dark object
- Classification of a dark object as oil pollution or natural slick

After highlighting the area of a dark object in the image, a number of its geometric and physical characteristics are calculated. A number of works [3-5] offer various characteristics of a dark object in radar images:

- Radar contrast background
- The ratio of the pixel intensity of the object / background
- The ratio of standard deviations of the object / background
- Entropy of contrast
- Contrast correlation
- Radar contrast of a dark object
- Medium contrast

- Standard deviation of the background
- Contrast
- The difference between the contrasts of the background and the object.

If the stain is the result of a recent oil spill by a moving vessel (for example, a tanker cleans its tank), an important pollution parameter is tensile, which can be expressed as a length / width ratio.

In [6], it was shown that the parameters of a dark object related to the intensity contrast gradient in the direction from the background to the object provide the greatest reliability of information if classification using neural networks is used. The standard deviation of the background takes into account the effect of wind speed. If the pixel intensity is compared with the reflected signal, information about the spatial correlation of neighboring pixels provides such a parameter as the texture of the image.

The following parameters of the dark image object were selected as input parameters of the neural network classifier:

- The size of the area on which the object is observed (in sq. km) A .
- The perimeter of P – the length (in km.) of the boundaries of the object.
- Complexity defined as $C = \frac{P}{2\sqrt{\pi A}}$. This parameter usually takes small numerical values for areas with simple geometry and large values for complex geometric areas.
- Length S . This parameter was obtained using the method of principal components [7] of vectors, the components of which are the coordinates of the pixels belonging to the object. If λ_1 and λ_2 are two eigenvalues associated with the calculated covariance matrix and $\lambda_1 > \lambda_2$, length value is $S = \frac{100\lambda_2}{\lambda_1 + \lambda_2}$.
- The standard deviation for the object (OSd) is the standard deviation (in dB) of the intensity values of pixels belonging to the dark object.
- Standard deviation for the background (BSd) – standard deviation (in dB) of the intensity values of pixels belonging to the area surrounding the dark object.
- Maximum contrast (ConMax) – the difference (in dB) between the average value of the intensity of the pixels in the background and the smallest value of the intensity of the pixels outside the dark object.
- Average contrast (ConMe) – the difference (in dB) between the average value of the intensity of the pixels in the background and the average value of the intensities of the pixels of a dark object.
- Maximum gradient (GMax) – maximum boundary gradient (in dB) “background-object”.
- Average gradient (GMe) – average boundary gradient (in dB) of the “background object”.
- Gradient standard deviation (GSd) – standard deviation in dB of the boundary gradient values.

The main statistical parameters that describe the objects of observation are shown in Table 1.

Table 1. The main statistical parameters that describe the objects of observation

Parameter	Oil slick				Nature slick			
	Min	Max	Avg	Statistical deviation	Min	Max	Avg	Statistical deviation
<i>A</i> , km ²	0,4	40,6	6,4	7,5	1,1	115,6	13,3	17,0
<i>P</i> , km	4,2	117,6	28,1	22,1	7,1	396,4	52,4	57,2
<i>C</i>	1,1	6,8	3,2	1,1	1,1	10,4	3,9	1,7
<i>S</i>	0,1	40,8	4,2	7,4	0,1	45,2	11,8	11,4
Osd, dB	0,8	3,8	1,7	0,6	0,9	3,2	2,0	0,6
BSd, dB	0,8	3,0	1,1	0,4	0,9	2,3	1,5	0,4
ConMax, dB	3,2	15,7	9,2	2,6	2,6	14,9	10,8	2,2
ConMe, dB	-0,4	10,9	4,8	1,9	-0,4	9,3	5,3	1,7
GMax, dB	2,8	15,5	7,2	2,2	3,6	16,8	8,5	2,6
GMe, dB	0,0	6,5	3,0	1,1	0,0	5,2	2,7	1,0
GSd, dB	0,8	2,7	1,4	0,4	0,6	2,6	1,5	0,5

The obtained values of the object parameters are used as a source vector for the neural network classifier of image objects.

3 Methodology for Determining the Structure of a Neural Classifier

Classification algorithms are mainly based on Bayesian or statistical solutions. The disadvantage of these methods is the difficulty in developing classification rules due to the involvement of many nonlinear and poorly studied factors in this process. These difficulties can be overcome by using neural network algorithms. Neural networks, unlike statistical classifiers, do not require a precisely defined relationship between input and output vectors, because form their own input-output relations from the data set, using the construction of the boundaries of solutions [8].

The main element for constructing the NN is an artificial neuron. It is traditionally represented as a linear sum with N inputs (a weight coefficient is assigned to each of the inputs) and one output connected to a nonlinear element that implements the activation function of a neuron.

When choosing the structure of a neural network to solve the classification problem, the following aspects should be considered:

- The ability of the network to learn, i.e. the ability to teach the system to recognize the required number of objects. The more layers and neurons in the network, the higher its abilities and, at the same time, the need for hardware resources.
- The speed that is achieved by reducing the complexity of the network, because the less hardware resources are needed, the faster the NN operation is.

Satisfying these conflicting requirements requires solving the problem of optimizing the structure of the NN. To solve this problem, you can use the Hegt-Nielsen theorem [9], which proves the representability of the function of many variables of a general

form using a two-layer direct propagation NN with limited activation functions of a sigmoid form:

$$F(x) = 1 / (1 + \exp(-x)). \quad (1)$$

Sigmoidal functions are monotonically increasing and have non-zero derivatives in the entire domain of definition. These characteristics ensure the proper functioning and training of the network [10],[11].

A two-layer NN of direct propagation having N inputs and K outputs contains a hidden layer consisting of M neurons and an output one consisting of K neurons. The result of the work of the NN is a nonlinear transformation of the N -dimensional input vector \mathbf{X} into the K -dimensional output vector \mathbf{Y} :

$$\mathbf{Y} = F((F(\mathbf{X} \cdot \mathbf{W})) \cdot \mathbf{V}) \quad (2)$$

where \mathbf{W} – is matrix of weights of the inner layer of NN of size $N \times M$;
 \mathbf{V} – is matrix of weights of the output layer of NN of size $M \times K$;
 F – activation function.

Expression (2) is the matrix form of the system of K nonlinear equations. By selecting variable values \mathbf{W} and \mathbf{V} it is possible to achieve approximate equality of value \mathbf{Y}_i to one of the predefined values \mathbf{Y}_i , mapped to one or more input vectors \mathbf{X}_i . This process is called network training. Iterations of NN training are repeated until, for all reference images, the required signal values at the NN outputs are reached with an error less than some predetermined learning error δ . Thus, the probability of a correct classification depends on a given value δ and on the number of neurons in each NN layer.

To reduce the complexity of the network, it is proposed to enter, not the pixel intensity values, but the previously obtained values of the characteristics of the image segments calculated at the previous stage as an input data array.

Thus, in the created neural network there are 11 input parameters and it is necessary to determine the number of neurons in the hidden layer. For the task of extracting information from inputs in order to generalize or reduce the dimension of the data array, it is necessary to use a narrowing network. The number of training examples should be approximately equal to the number of network weights multiplied by the reciprocal of the error. For example, for a marginal error $\varepsilon=0.1$ it is necessary to use a training data set 10 times larger than the number of weights. This dependence is described by the formula:

$$n \geq \frac{\omega}{\varepsilon}, \quad (3)$$

where n – training dataset volume;

ω – number of weights in the network.

The reason that the magnitude of the error plays a significant role is related to the relationship between the generalizing ability and accuracy. A small mistake in a re-trained network cannot be considered a training success. If you need to use a larger number of weights than the one that can fill the data set, then you need to stop with a

larger learning error in order to maintain a generalizing ability. This forces sacrifice of accuracy in favor of the generalizing ability of the network.

Since the number of input and output elements in most cases is determined by the task, we can define an expression that describes the number of weights in terms of the number of hidden units in a fully connected unidirectional network with one hidden layer:

$$h = \frac{\omega}{io}, \quad (4)$$

where i – number of input network variables;
 o – number of output network variables;
 h – number of neurons in the hidden layer.

Thus, we get that in a neural network with one hidden layer, eleven input variables and one output, provided that the number of neurons in the hidden layer $h \leq 10$ the number of weights must be at least 110.

4 Algorithm for Testing the Suitability of the Selected Structure of a Neural Network for Solving the Classification Problem

To test the suitability of the selected neural network structure for solving the classification problem, we consider a process characterized by the desired value y_p at the output of a neural network, which is one of the possible implementations of a random variable \mathbf{Y} . The expected value of the output variable \mathbf{Y} can be expressed as a function of the vector of input parameters of a neural network x in the following way:

$$Y = \mu(x) + W, \quad (5)$$

where μ – regression function;
 W – random variable with zero mean and final variance.

Let there be a set of training data that can be repeated. Repetitions provide a neural network model-independent estimate of noise variance σ^2 . Denote by M the number of different sets of values of the input variables of the neural network, and N_k - the number of repetitions of the vector of input values of the neural network x^k , $k = \overline{1, M}$. Then $y_p^{k,j}$ denotes the value of the output process for the j -th repetition of the input vector

x^k , and $\bar{y}^k = \frac{1}{N_i} \sum_{j=1}^{N_i} y_p^{i,j}$ – the average output process at x^k . Total network training dataset size $N = \sum_{k=1}^M N_k$.

Consider the orthogonal vectors $\{e_i\}_{i=1, \overline{M}}$ in Euclidean space, such that the first N_1 values of first vector \bar{e}_1 are equal to one, and the rest are zero.

Components from N_1+1 to N_1+N_2 of vector \vec{e}_2 are equal to one, and the rest are zero. Last N_M values of components of vector $\{\vec{e}_i\}_M$ are equal to one, and the rest are zero.

Then the value at the output of the neural network for all M sets of values of the input variables $y_{pM} = \sum_{i=1}^M \bar{y}_p^i e_i$. The vector of the difference between the desired and actual value at the output of the neural network $r = y_p - y$, where y – the value at the output of a neural network that has passed the procedure for selecting weights (parameters), i.e. at the output of the model. Net output error vector $p = y_p - y_{pM}$. Model error vector is an indicator of its displacement $m = y_{pM} - y$.

The sum of the squared differences of the neural network model can be written as

$$r^T r = p^T p + m^T m = \sum_{k=1}^M \sum_{j=1}^{N_k} (y_p^{k,j} - \bar{y}_p^k)^2 + \sum_{k=1}^M N_k (y_k - \bar{y}_p^k)^2. \quad (6)$$

If the value $\frac{m^T m}{(M-q)}$, where q – the number of weights is large enough compared to the value $\frac{p^T p}{(N-M)}$, then this neural network model is not suitable for solving the problem.

Let the null hypothesis H_0 consists in the fact that the model is unbiased, i.e. the family of functions determined by the structure of the neural network contains a regression. If hypothesis H_0 is true, then attitude $u = \frac{m^T m}{M-q} / \frac{p^T p}{N-M}$ is the value of a random variable distributed according to Fisher's law with $(M-q)$ and $(N-M)$ degrees of freedom. Decision to reject the hypothesis H_0 (model does not fit) with risk $\alpha\%$, if hypothesis H_0 is true (error of the first kind) is accepted if $u > f_{N-M}^{M-q}(1-\alpha)$, where f_{N-M}^{M-q} is the inverse of the cumulative Fisher distribution. If $u \leq f_{N-M}^{M-q}(1-\alpha)$ we can say that the selected model is unbiased, i.e. contains a regression.

To evaluate weights (parameters θ) direct distribution neural networks often use the least squares method, which consists in the selection of weights in which a minimum cost function is achieved

$$F(\theta) = \frac{1}{2} \sum_{k=1}^N (y_p^k - f(x^k, \theta))^2. \quad (7)$$

The cost function is minimized using iterative algorithms, for example, using the Levenberg-Marquart algorithm. At the i -th iteration, the vector of network parameters in the previous step is known θ_{i-1} , and the parameter vector at the current step is calculated using the formula:

$$\theta_i = \theta_{i-1} + (z_i^T z_i + \lambda_i I_q)^{-1} z_i^T (y_p - f(x, \theta_{i-1})), \quad (8)$$

where $z_i = \left. \frac{df(x, \theta)}{d\theta^T} \right|_{\theta=\theta_{i-1}}$;

$[z_i]_{kj} = \left. \frac{df(x^k, \theta)}{d\theta_j} \right|_{\theta=\theta_{i-1}}$ – Jacobian matrix at the i -th iteration;

i_q – unit matrix.

Since the Jacobian is a matrix of size $M \times q$, then its calculation is a rather costly procedure for time and computer memory, however, in [12] a faster method for calculating the Jacobian is proposed, which does not affect the continuity of the function $f(x, \theta)$, consisting in the fact that the derivative is calculated immediately over the entire range of changes in the input value x^k .

When the scalar λ vanishes, formula (8) degenerates into an increment formula for the Newton method, using the approximated Hessian matrix. When λ is different from zero, the gradient begins to decline in small steps. At very large values λ we get the steepest descent method. The Levenberg-Marquart method is faster and more accurate in the region of the minimum error; it also seeks to increase the speed of learning as much as possible. In this way, λ decreases after each satisfactory step (decrease in the presented function) and increases only when the minimized function increases in the preliminary step. Thus, the minimized function will always decrease with each step of the algorithm.

To achieve a minimum of the cost function, several minimization procedures must be performed with different initial conditions [13].

Rating $(1-\alpha)\%$ of confidence interval for regression for any desired value at the input of the network x^a is given by function

$$f(x^a, \theta_{LS}) \pm t_{N-q} \left(1 - \frac{\alpha}{2}\right) s \sqrt{(z^a)^T (z^T z)^{-1} z^a}, \quad (9)$$

where t_{N-q} – is function inverse of student distribution with $(N-q)$ degrees of freedom.

If N is large, then the Student distribution is close to Gaussian. $s \sqrt{(z^a)^T (z^T z)^{-1} z^a}$ – is standard deviation of the least-squares regression function.

Inappropriate models of a neural network can be eliminated by estimating the Jacobian matrix z . A poorly conditioned z matrix is a symptom that some of the network weights are useless, that the neural network model is too complex or overridden. It is proposed to evaluate the matrix z by calculating the ratio $k(z)$ – its largest and smallest singular number. The matrix z can be considered poorly defined if the value $k(z^T z) = k^2(z) \leq 10^8$. Moreover, the value $k(z)$ increases with increasing number of neurons in the hidden layer.

A useful approximation to the k -th single error is the expression

$$e^k \approx \frac{r^k}{1 - [p_z]_{kk}}; \quad k = \overline{1, N}, \quad (1)$$

where $p_z = z(z^T z)^{-1} z^T$ denotes an orthogonal projection matrix of size $N \times N$ on the Jacobian matrix z of size $N \times q$.

The approximate value of many individual errors $Ap = \frac{1}{N} \sum_{k=1}^N (e^k)^2$ can only be calculated if the matrix $z^T z$ is well defined.

We can say that the neural network model contains a good regression estimate if the ratio Ap to her standard error of training $MSTE = \frac{1}{N} \sum_{k=1}^N (y_p^k - f(x^k, \theta_{LS}))^2$ is close to one.

5 Results

To implement the algorithm for classifying monochrome images using neural networks, a software product has been created that can be used in remote monitoring systems to obtain real-time information about the state of pollution of port and coastal areas [14].

The probability of the presence of oil pollution P , issued by an program based on the developed neural network, is interpreted in accordance with confidence levels:

- $P < 0,3$ – low confidence;
- $0,3 < P < 0,9$ – average confidence level;
- $P > 0,9$ – high confidence level.

A high level of confidence is accepted when:

- a dark object has a high contrast compared to the surrounding background;
- the surrounding background is homogeneous;
- wind speed is in the range from 6 to 10 m/s;
- oil tanker or drilling platform directly connected to a dark object.

The average level of confidence is accepted when:

- wind speed is in the range from 3 to 6 m/s;
- a dark object has a low contrast to the gray level that determines the surrounding background, especially at wind speeds from 6 to 10 m/s;
- the shape of the dark object is asymmetric, i.e. rough edges.

A low level of confidence is accepted when:

- a dark object is in the area of low-speed winds;
- natural slicks found near a dark object;
- a dark object has a low contrast to the gray level that defines the surrounding background.

Using the developed algorithm, several anomalies of the sea surface were revealed, shown in Fig. 2 and Fig. 3.

Fig. 2 shows the image where a zone with dark stripes is marked. Dark areas are caused by areas of small winds, as a result of which the sea surface becomes less rough

and mirrors all incident radiations. In the lower part of the image, at the base of the Curonian Spit, anomalies caused by natural films are clearly visible. Natural films are formed as a result of the life of plankton and fish. They are very useful in detecting the circulation of surface currents, but they are easily destroyed at a wind speed of more than 7 m/s.



Fig. 2. Identified anomaly "natural film"

Figure 3 shows the image of the eastern part of the Gulf of Finland and Lake Ladoga. The white area covering most of the lake is classified as not melted ice.

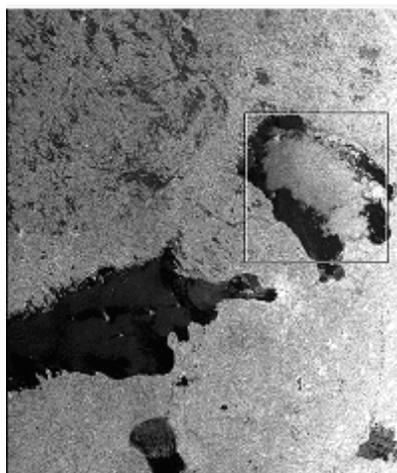


Fig. 3. Identified anomaly "not melted ice"

6 Conclusion

In the presented design method of the neural network classifier of oil pollution on the water surface, the input parameters of the neural network, the structure of the neural classifier are determined. and an algorithm is proposed for testing the selected structure of the neural network for suitability for classifying the dark area in the image of the water surface as oil pollution or wind slick.

The algorithm for creating the classifier of image objects based on a neural network implemented programmatically. The program of a neural network classifier of objects in images allows improving the quality of environmental pollution detection characteristics..

References

1. Richards, J.A.: Remote sensing with Imaging Radar. Springer, New York (2009).
2. Brekke, C. and Solberg, A. H. S.: Oil Spill Detection by Satellite Remote Sensing. *Remote Sensing of Environment* 95(1), 1-13 (2005).
3. Solberg, A. H. S., Storvik, G., Solberg, R. and Volden, E.: Automatic Detection of Oil Spills in ERS SAR Images. *IEEE Transactions on Geoscience and Remote Sensing* 37(4), 1916-1924 (1999).
4. Fiscella, B., Giancaspro, A., Nirchio, F., Pavese, P. and Trivero, P.: Oil spill detection using marine SAR images. *International Journal of Remote Sensing* 21(18), 3561-3566 (2000).
5. Topouzelis, K., Karathanassi, V., Pavlakis P. and Rokos, D.: Oil Spill Detection: SAR Multiscale Segmentation and Object Features Evaluation. In *Proceedings Remote Sensing of the Ocean and Sea Ice*, vol. 4880, pp. 77-78 (2002).
6. Marghany, M.: RADARSAT Automatic Algorithms for Detecting Coastal Oil Spill Pollution. *International Journal of Applied Earth Observation and Geoinformation*, 3(2), 191-196 (2001).
7. Assilzadeh, H., Mansor, S. B.: Early Warning System for Oil Spill Using SAR Images. In *22nd Asian Conference on Remote Sensing (ACRS 2001)*, vol 1, pp. 460-465. Singapore (2001).
8. Rahnama M., Gloaguen R. *TecLines: A MATLAB-based toolbox for tectonic lineament analysis from satellite images and DEMs. Pt 1: Line segment detection and extraction. Remote Sensing*, 6(7), 5938–5958 (2014).
9. Krishnakumar, K., Ganesh Karthikeyan, V.: Analysis and performance evaluation of large data sets using Hadoop. *International Journal of Research in Advent Technology* 2(5), 245-250 (2014).
10. Haykin, S.S.: *Neural Networks and Learning Machines*. PHIL Publ., Canada (2010).
11. Sovetov, B. Y., Tatarnikova, T. M. and Cehanovsky, V. V.: Detection System for Threats of the Presence of Hazardous Substance in the Environment. In *XXII International Conference on Soft Computing and Measurements (SCM)*, pp. 121-124. St. Petersburg, Russia (2019). doi: 10.1109/SCM.2019.8903771.
12. Wilanowski, B.M., Iplikci, S., Kaynak, O, Efe, O. M.: An algorithm for Fast Convergence in Training Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1778-1782. Washington, DC, USA (2001).

12 T. Tatarnikova, E. Chernetsova

13. Kutuzov, O. I. and Tatarnikova, T. M.: On the Acceleration of Simulation Modeling. In: XXII International Conference on Soft Computing and Measurements (SCM), pp. 45-47. St. Petersburg, Russia (2019). doi: 10.1109/SCM.2019.8903785.
14. Bogatyrev, V.A., Bogatyrev, S.V., Derkach, A.N.: Timeliness of the Reserved Maintenance by Duplicated Computers of Heterogeneous Delay-Critical Stream. In CEUR Workshop Proceedings, vol. 2522, pp. 26-36 (2019).