Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

XIIIth International Symposium "Intelligent Systems", INTELS'18, 22-24 October 2018, St. Petersburg, Russia

All convolutional neural networks for radar-based precipitation nowcasting

G. Ayzel[a,b,*], M. Heistermann[a], A. Sorokin[c], O. Nikitin[c], O. Lukyanova[c]

[a] *Institute of Earth and Environmental Science, University of Potsdam, Potsdam, 14476, Germany*
[b] *Institute of Water Problems of Russian Academy of Sciences, Moscow, 119333, Russia*
[c] *Computing Center Far Eastern Branch Russian Academy of Sciences, Khabarovsk, 680000, Russia*

Abstract

Today deep learning is taking its rise in hydrometeorological applications, and it is critical to extensively evaluate its prediction performance and robustness. In our study, we use deep all convolutional neural networks for radar-based precipitation nowcasting, which has a crucial role for early warning of hazardous events at small spatiotemporal scales. Our trial and error study focuses the particular importance of selecting and adopting suitable data preprocessing routine, network structure, and loss function regarding input data features, and, as a result, highlights limited transferability of methods in existing studies. Results show that parsimonious deep learning models can forecast a complex nature of a short-term precipitation field evolution and compete for the state-of-the-art performance with well-established nowcasting models based on optical flow techniques.

Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems".

Keywords: precipitation; nowcasting; convolutional networks; deep learning.

1. Introduction

Precipitation is a crucial driver for many natural hazardous events (landslides, flash-floods), and its early prediction plays a crucial role in developing new warning systems. If operationalized, such systems can substantially reduce economic and human losses caused by hazardous events [1,2]. However, the field of radar-based precipitation nowcasting is not well-developed. For the last 20 years this field went not so far from the standard workflow which utilizes an implementation of optical flow technique as a core of methodology [3], and then coupled with numerical weather predictions (NWP) [4] and/or stochastic field perturbations [4,1].

The first papers which have started the new methodological direction in the field of radar-based precipitation nowcasting were introduced by the group from Hong-Kong Meteorological observatory in 2015 [5] and 2017 [6]. In the 2015-th paper, authors introduced ConvLSTM model and showed that it overperforms current operational SWIRLS system for radar-based precipitation nowcasting for lead times up to two hours. In the 2017-th paper, the same authors introduced TrajGRU model which utilizes location-variant transformations (in comparison with location-invariant

* Corresponding author. Tel.: +49-331-977-6454.
E-mail address: ayzel@uni-potsdam.de

1877-0509
Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems".

convLSTM) and overperforms both SWIRLS and convLSTM models. Despite their novelty, these papers are going to be task-specific (have a limited potential for spreading over the community) because of complicated and area-specific procedure of radar data preprocessing (smoothing, removing outliers, scaling) and non-transparent development of the proposed networks' architecture. As far as we know, there are the only approaches in the field to tackle precipitation nowcasting with deep learning models.

This paper aims to provide a simplified and transparent approach to developing a research workflow around a precipitation nowcasting problem based on using deep learning models. We establish a step-by-step hypothesis testing about an impact of data preprocessing, loss function, and convolutional kernel size on a performance of a parsimonious all convolutional net. The best candidate model will be used as a baseline for further work.

The paper is organized as follows. In Section 2 we describe the data we used. Section 3 describes proposed models' architecture and features, and also introduces the experiment design, verification procedure, and research software we use. We report obtained results and discuss them in various contexts in Section 4. Section 5 provides an overall summary and conclusions.

2. Data

We use the so-called RY product of the German Weather Service (DWD) as input to our nowcasting models. The RY product represents a quality-controlled rainfall depth product (in mm) that is composited from the 18 Doppler radars operated by the DWD. The RY product has a spatial extent of 900×900 km and covers the whole territory of Germany. Temporal and spatial resolution of the RY product is 1×1 km, and 5 min, respectively. Data is available for the period from 2006 to 2017 and has 2013 year missing.

It is common practice in the machine and deep learning fields to perform some normalization steps before feeding the data to a neural network[7]. Unfortunately, a standard data normalization workflow for images as an input data does not fit our needs because of the different nature of photography images (RGB channels, 8-bit integers) and radar images (one channel, 64-bit floats). In this work we tried to perform four different transformation techniques, based on the RY data distribution features:

1. ”Scaler v.1” performs transformation proposed by Woo et. al[2] with the following parameters: threshold value = 1, normalization constant = 0.5.
2. ”Scaler v.2” performs transformation proposed by Woo et. al[2] with the following parameters: threshold value = 0.5, normalization constant = 0.25.
3. ”Scaler v.3” takes natural logarithm from input values added by 1.
4. ”Scaler v.4” takes natural logarithm from input values added by 0.01.

As of more than 95% of the RY data lie between 0 and 1 (mm of rainfall depth), the general idea of the implemented converters is to pay more attention for the dominant class and make it more deployed for a neural network. Figure 1 represents implemented transformations.

3. All convolutional neural networks

3.1. Architecture and features

All convolutional neural nets were introduced by Springenberg[8] and utilize an idea to replace pooling layers in deep neural networks by sequences of convolutional layers with increased strides. These type of architecture was also proposed by Mathieu[9] as a first naive approach for the next frame video prediction task. We propose a simple all convolutional neural net architecture (DozhdyaNet) with six subsequent 2D convolutional layers parametrized only by kernel size used (Fig. 2).

In DozdyaNet we do not utilize strides and always use zero padding in convolutional layers' structure for preserving the spatial resolution of radar image for a whole net. For providing the DozhdyaNet nonlinear flexibility, we use ReLu activation function[10] followed by every convolutional layer (except the last one). Parameters of the proposed

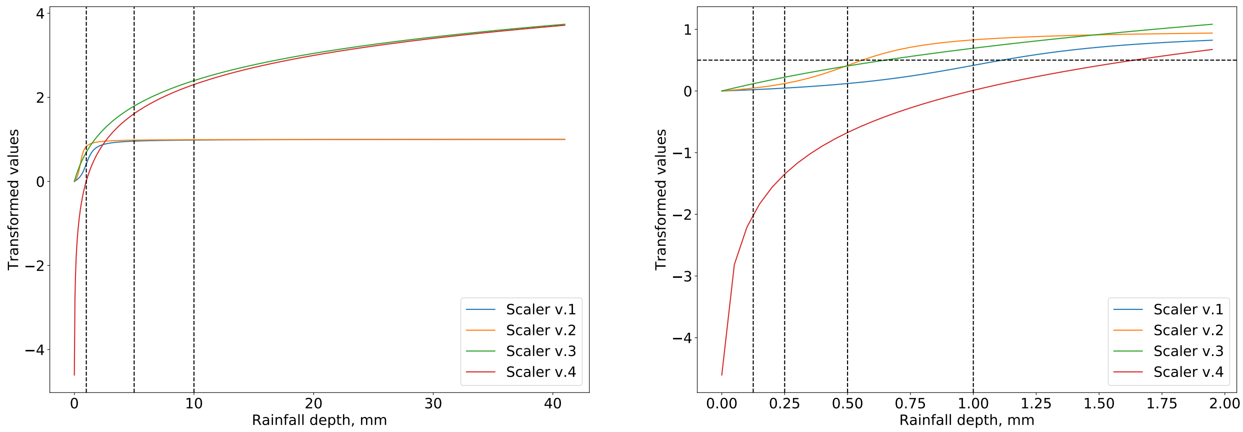

Fig. 1. Different transformations for mapping rainfall depth values (0-41 mm) to neural network input data

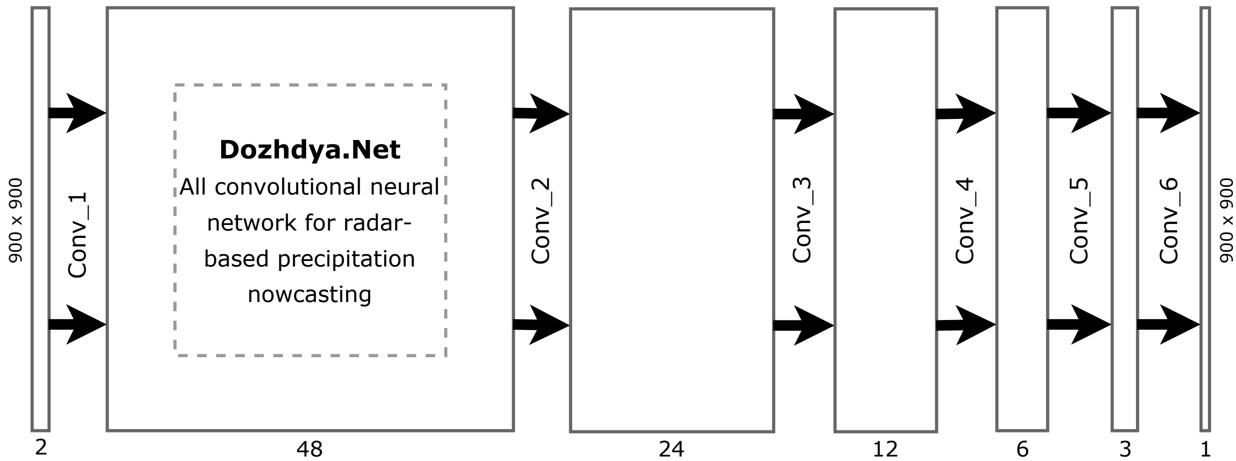

Fig. 2. All convolutional neural net baseline architecture

architecture are the number of filters and size of a convolutional kernel used in every layer. We propose to use filters' number of 48, 24, 12, 6, 3, and 1 from the first to the last layers in the DozhdyaNet, respectively, as a baseline.

Two last consequent radar images ($t-2, t-1$) serve as input for our models, and the model's target output is the radar image of the t-th time step. For one hour in advance predictions we make a nowcast consequently for every 12 time horizons (inside an hour), using the last predicted radar image as input for following calculations. This setting declines the number of model parameters and significantly reduces a computational time in comparison with the approach utilizes neural network structure for predicting every nowcast time horizon simultaneously.

3.2. Experiment design

One of the primary goals of this study is to find a starting point in a development deep neural networks in the field of radar-based precipitation nowcasting. Thus, to test different preprocessing routines on candidate models' performance (see Section 2) we also develop another two hypotheses for evaluation: how do a kernel size and a loss function impact all convolutional net performance?

We tried to use different sizes of convolutional kernels to test the influence of increasing receptive field on network's performance: 3×3, 5×5, 7×7.

Despite the rapid growth of deep learning field, there are also no clear guidelines which loss function do we need for its utilization in training process for our problem of vector regression and this highly sparse data. In this work we also tried to experiment with different loss functions: mean absolute error (MAE), mean squared error (MSE), logcosh function (Logcosh).

We train our candidate models on the RY data from 2006 to 2015 and use 2016 and 2017 for testing. In this study we use "Adam" algorithm with the default parameters proposed in the corresponding paper[11] for neural network weights' optimization. Training process utilizes mini-batches of random input data samples of size 12 and number of training epochs is 5. Training had been running on Nvidia Tesla P100 GPUs, where one training epoch took near 2 hours in a "single GPU" mode.

After finding the best candidate model in the space of different scalers, kernel sizes, and loss functions variants, we fine tune this model with increased training epochs and will use it as a baseline model of this type of proposed architecture (Dozhdya.Net v.1.0).

3.3. Verification procedure

For the analysis, we selected the "Braunsbach event" – the hazardous event which took place in Braunsbach town on 29 May 2016 and was triggered by torrential rainfall[12].

For the verification procedure, we use two general categories of scores: continuous (based on the differences between nowcast and observed rainfall intensities) and categorical (based on standard contingency tables for calculating matches between boolean values which reflect the exceedance of specific rainfall intensity thresholds). We use the mean absolute error (MAE) as a continuous score:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Now_i - Obs_i| \tag{1}$$

where Now_i and Obs_i are nowcast and observed rainfall rate in the i-th pixel of the corresponding radar images, and n the number of rainfall pixels.

And we use critical success index (CSI) as a categorical score:

$$CSI = \frac{hits}{hits + falsealarms + misses} \tag{2}$$

where $hits$, $misses$, and $falsealarms$ are are defined by the contingency table (Table 1) and the corresponding threshold value (1 mm/h in our study).

Table 1. Contingency table.

Nowcast	Observation (Yes)	Obseration (No)
Yes	hit	false alarm
No	miss	correct negative

3.4. Research software

In our research we use only open and freely available software libraries in Python programming language: wradlib (wradlib.org[13]), h5py (h5py.org[14]), Matplotlib (matplotlib.org[15]), TensorFlow (tensorflow.org[16]), NumPy (numpy.org[17]), and Keras (keras.io[18]).

4. Results and discussion

We have summarized all the impacts of tested hypothesis on Fig. 3.

4.1. Data preprocessing impact

It is clear that Scaler v.4 which utilizes natural logarithm applied to our rainfall depth values with small addition (0.01, for avoiding null values) provides better results regarding both MAE and CSI scores. This result can be explained by a particularly equal deployment of depth values under 1 mm, and over 1 mm to transformed values scale, which provides a high attention to rainfall depth values we are most interested in. Thus this type of Scaler is of high universality; it can be recommended for using as a first guess.

4.2. Loss function impact

Our results show that a choice of a loss function is of critical importance for nowcasting study. The most widely used for regression problems, MSE function, showed significantly worse results in our setting in comparison with Logcosh function. This result can be explained by the Logcosh function features which utilize both advantages of using squared (MSE) loss for small values and absolute (MAE) loss for large values. This natural conditional approach works here because of it enhances model's universality, which is important for using one model for very different rainfall events.

4.3. Kernel size impact

Our first guess was that increasing the neural network receptive field by increasing a size of convolutional kernels, would increase the overall network's performance because of accounting of precipitation specific changes from a larger neighborhood. However, the results are different. It is clear that increasing model's receptive field capacity leads to overfitting and increasing uncertainty of predictions. For providing nowcast with the lead time of one hour, we recommend using smaller kernel size of convolutional layers.

4.4. Comparison with optical flow and persistence models

Based on obtained results, we defined the first major version of Dozhdya.Net model (v.1.0) with the following parameters: convolutional kernel sizes of 3×3, Logcosh loss function, and the last scaler version (Scaler v.4) for data preprocessing. Fig. 4 shows the reached performance in comparison with (weak) Eulerian persistence models and state-of-the-art performance of the global optical flow based model.

Results show that our parsimonious baseline model which was build based on a clear workflow and computationally cheap experimentation with different hypotheses about model's features significantly overperforms Eulerian persistence model and can compete with the state-of-the-art global optical flow based model. Theoretically, there is a distinct advantage and disadvantage of deep learning approaches under optical flow. The disadvantage is that such simple all convolutional architectures are location-invariant, so they provide the same transformations of rainfall field for every radar image that leads to ignoring geographical features of rainfall patterns. The advantage is that deep learning models can directly take into consideration growth and decay processes of rainfall intensity evolution during an event. There is much room for experimentation, and here we provide a solid baseline and starting point for further development and evaluation results improvement.

5. Conclusions

We evaluated an impact of different data preprocessing routines, loss functions, and convolutional kernel sizes on an evaluation performance of proposed deep learning models. Considering the results of this experimentation, we introduced a baseline model (Dozhdya.Net v.1.0) for radar-based precipitation nowcasting based on all convolutional neural network architecture. This parsimonious model shows a comparable efficiency in comparison with the state-of-the-art optical flow based model and can be used as a solid baseline for further development.

We also want to underline a critical role of open source software, computational facilities, and community help for developing such a new field in geoscientific modeling. Only being concerted these building blocks of modern science can lead to new results.

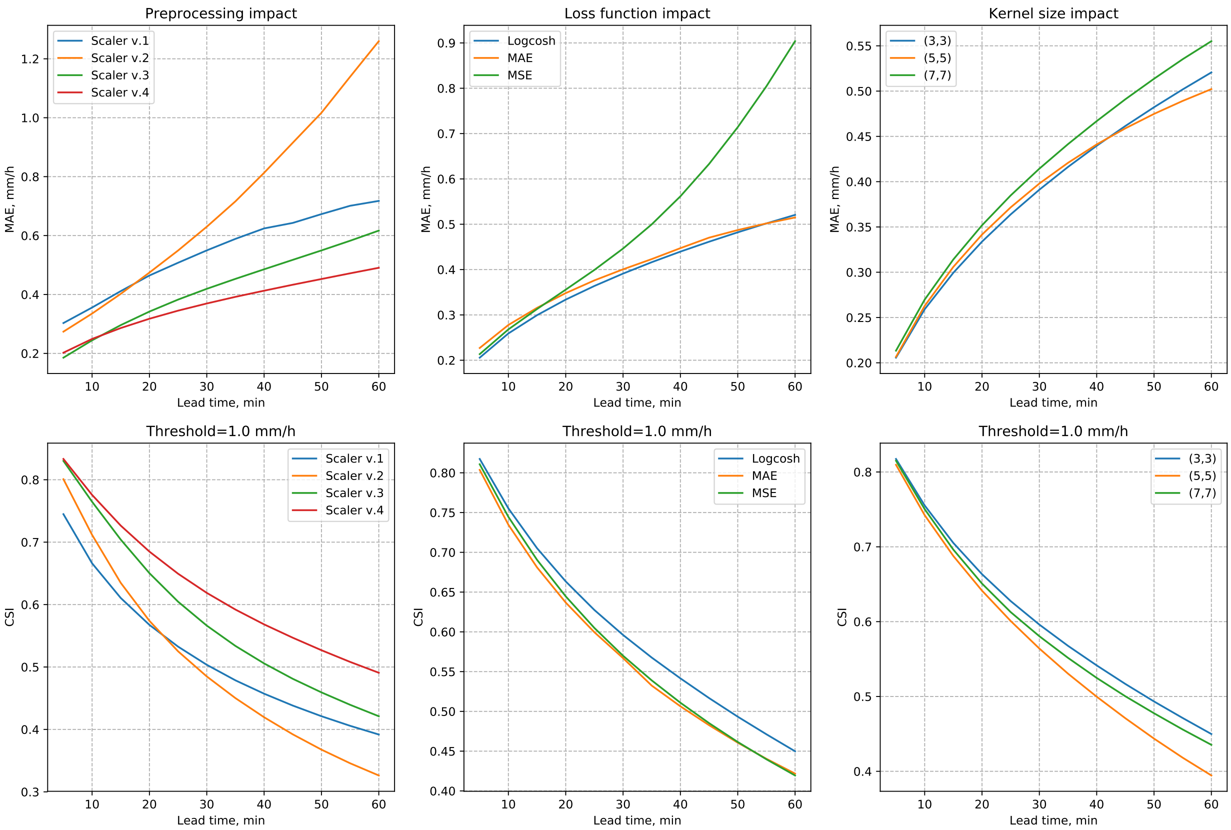

Fig. 3. Parameters' impact on the verification period performance

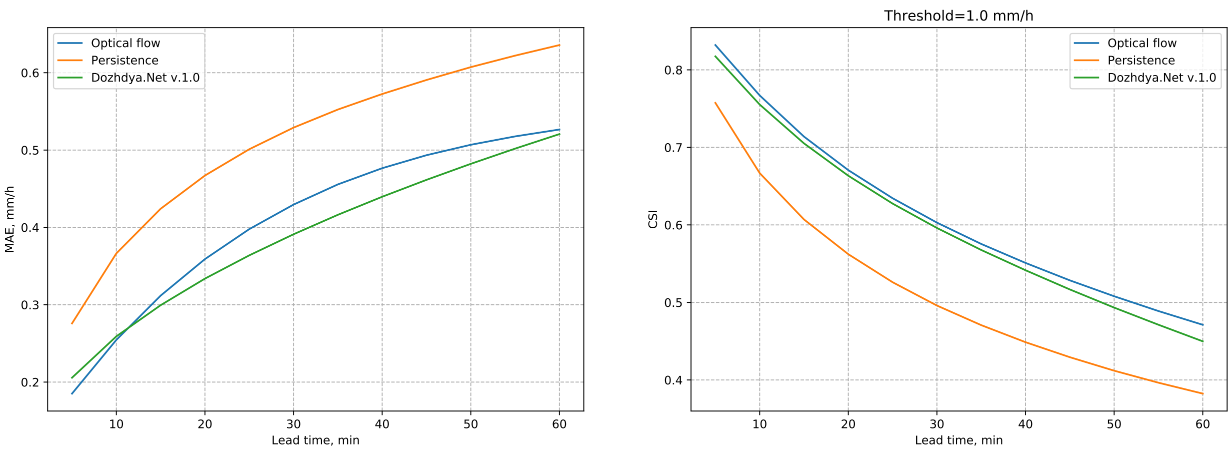

Fig. 4. Comparison of the Dozdya.Net baseline architecture with global optical flow and Eulerian persistence models

Our future work will be devoted to developing new and universal deep learning architectures which can be easily transferred between different study areas and radar products.

Acknowledgements

This publication was financially supported by Geo.X, the Research Network for Geosciences in Berlin and Potsdam (Project-number: SO_087_GeoX). Computations were performed with the methods and techniques, which were developed under the RFBR scientific project number 18-29-03196. We run our experiments using GPU computation resources of the Shared Facility Center Data Center of FEB RAS (Khabarovsk). The authors would like to thank Open Data Science community (ods.ai) for many valuable discussions and educational help in the growing field of machine/deep learning.

References

1. Foresti, L., Reyniers, M., Seed, A., Delobbe, L.. Development and verification of a real-time stochastic precipitation nowcasting system for urban hydrology in belgium. *Hydrology and Earth System Sciences* 2016;**20**(1):505–527. URL: `https://www.hydrol-earth-syst-sci.net/20/505/2016/`. doi:10.5194/hess-20-505-2016.
2. Woo, W.c., Wong, W.k.. Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting. *Atmosphere* 2017;**8**(3).
3. Bowler, N.E., Pierce, C.E., Seed, A.. Development of a precipitation nowcasting algorithm based upon optical flow techniques. *Journal of Hydrology* 2004;**288**(1-2):74–91.
4. Bowler, N.E., Pierce, C.E., Seed, A.W.. Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled nwp. *Quarterly Journal of the Royal Meteorological Society* 2006;**132**(620):2127–2155.
5. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting 2015;URL: `http://arxiv.org/abs/1506.04214`.
6. Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.Y., Wong, W.k., et al. Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model 2017;URL: `http://arxiv.org/abs/1706.03458`.
7. Coates, A., Ng, A., Lee, H.. An analysis of single-layer networks in unsupervised feature learning. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011, p. 215–223.
8. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.. Striving for simplicity: The all convolutional net. *CoRR* 2014; **abs/1412.6806**. URL: `http://arxiv.org/abs/1412.6806`.
9. Mathieu, M., Couprie, C., LeCun, Y.. Deep multi-scale video prediction beyond mean square error. *CoRR* 2015;**abs/1511.05440**. URL: `http://arxiv.org/abs/1511.05440`.
10. Nair, V., Hinton, G.E.. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, p. 807–814.
11. Kingma, D.P., Ba, J.. Adam: A method for stochastic optimization. *CoRR* 2014;**abs/1412.6980**. URL: `http://arxiv.org/abs/1412.6980`.
12. Bronstert, A., Agarwal, A., Boessenkool, B., Crisologo, I., Fischer, M., Heistermann, M., et al. Forensic hydro-meteorological analysis of an extreme flash flood: The 2016-05-29 event in braunsbach, sw germany. *Science of The Total Environment* 2018;**630**:977 – 991. URL: `http://www.sciencedirect.com/science/article/pii/S0048969718306405`. doi:https://doi.org/10.1016/j.scitotenv.2018.02.241.
13. Heistermann, M., Jacobi, S., Pfaff, T.. Technical note: An open source library for processing weather radar data (wradlib). *Hydrology and Earth System Sciences* 2013;**17**(2):863–871. URL: `https://www.hydrol-earth-syst-sci.net/17/863/2013/`. doi:10.5194/hess-17-863-2013.
14. Collette, A.. *Python and HDF5: Unlocking Scientific Data*. " O'Reilly Media, Inc."; 2013.
15. Hunter, J.D.. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering* 2007;**9**(3):90–95. doi:10.1109/MCSE.2007.55.
16. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. URL: `https://www.tensorflow.org/`; software available from tensorflow.org.
17. Oliphant, T.E.. *A guide to NumPy*; vol. 1. Trelgol Publishing USA; 2006.
18. Chollet, F., et al. Keras. `https://keras.io`; 2015.