

# Using Boosted k-Nearest Neighbour Algorithm for Numerical Forecasting of Dangerous Convective Phenomena

E. N. Stankova(✉) and E. V. Khvatkov

Saint-Petersburg State University,
7-9 Universitetskaya nab., St. Petersburg 199034, Russia
e.stankova@spbu.ru, e.hvatkov@gmail.com

**Abstract.** The authors propose a boosted k-nearest neighbour algorithm for numerical forecasting of dangerous convective phenomena. The algorithm is applied for processing the output data of the numerical cloud model. The results show that boosted algorithm is able to predict the very fact of convective phenomena occurrence with the high accuracy, but it is not so good in distinguishing the specific type of convective phenomena. Comparison with the k-NN algorithm without boosting shows that boosting resulted in better accuracy of forecasting.

**Keywords:** Machine learning · Naive bayes classifier · Boosted k-nearest neighbour algorithm · AdaBoost algorithm · Support-vector machine model · Numerical model of convective cloud · Weather forecasting · Thunderstorm forecasting

## 1 Introduction

Constant anthropogenic influence on the atmosphere is resulted in global warming, which in turn leads to an increase in the intensity of convective processes. An increase in temperature and an increase in air humidity are two facts that together lead to the intensification of active convection in the atmosphere, which entails an increase in thunderstorm activity, in the number of heavy showers, tornadoes and other dangerous convective phenomena that have a tremendous destructive effect such as forest fires or flood waters. Therefore, the problem of timely prediction of dangerous convective phenomena is one of the most relevant and practically important areas of research.

The development of convection is determined by the physics of the complex interaction of dynamic, microphysical and electrical processes occurring in the cloud. Computer modeling is the most successful instrument for cloud research allowing to provide detailed investigation of variable specificity of convective clouds caused by high vertical velocities within the cloud and its surroundings without expensive field experiments. Computer research is based on numerical simulation, the result of which is a large amount of data that needs to be analyzed using modern methods of data analysis and processing.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 802–811, 2019.
https://doi.org/10.1007/978-3-030-24305-0_61

As a rule, modern models of convective clouds do not contain a block describing electrical processes due to complexity of the block realization and the large amount of required computing resources. In this regard, it seems relevant to investigate the possibility of forecasting the probability of such a dangerous convective phenomenon as a thunderstorm, only by the values of dynamic and microphysical parameters calculated in the process of modeling.

At present, machine learning seems to be the most adequate tool for such kind of research. The concept of machine learning implies processing the large amounts of data and identification various interrelations and patterns for the restoration of dependencies by empirical data.

The use of machine learning methods in meteorology is twofold. On the one hand, "pure" machine learning models are being developed, where certain atmospheric parameters are predicted on the basis of observational data obtained at meteorological stations, weather centers, etc. On the other hand, machine learning methods are used to verify models by establishing relationships between model forecasts and the actual meteorological situation. Both approaches were described in detail in our previous works [1–4] where we analyze the achievements of Microsoft [5], Yandex [6], and IBM [7] companies in the field of application of information technologies for weather forecasting.

Last year Yandex company used machine learning algorithms for the implementation of nowcasting technology, which is a technology of a short-term hyperlocal precipitation forecast [8].

Our previous works concerned the usage of machine learning technology both for validation of one and half dimensional numerical model of a convective cloud [1, 4] and for the operational forecast of dangerous convective phenomena using the verified model [2, 3]. Three methods of machine learning were used: Support Vector Machine (SVM), Logistic Regression (LR) and Ridge Regression (RR), providing rather high value of accuracy, up to 98%. But these methods, especially SVM, are rather expensive for training and have a lot of meta parameters for tuning. So now we decide to use k-Nearest Neighbour algorithm [9] which is, in opposite, cheap for training, but may be subject to retraining. We use AdaBoost algorithm [10] in conjunction with k-nearest neighbour algorithm (boosted k-NN) to avoid this drawback. Some preliminary results are presented in the present paper.

## 2 Data Selected for Research

Integrated information system [11–13] was applied for gathering of specific meteorological data used as the model input parameters. 480 radiosonde soundings were collected in the place and at the time when dangerous convective phenomena were observed, including 220 soundings for thunderstorms, 174 ones for heavy rains and 86 soundings for light rains. 196 soundings were collected in the cases when no dangerous convective phenomena were observed.

Aone and a half dimensional, time-dependent model with a detailed description of microphysical processes of convective cloud [14–18] was used to carry out numerical calculations. The model allows one to obtain dynamic and microphysical

characteristics of a convective cloud with a high degree of reliability without large computational costs. Model calculations were provided in automated regime in order to process series of input soundings at a time.

The output data of the model were grouped in sets, each of the set included 12 cloud parameters and the variable that described the type of dangerous convective phenomenon.Calculated cloud parameters were as follows:

- the vertical component of the velocity,
- radial component of the velocity,
- pressure,
- air density,
- ambient temperature,
- temperature excess in the cloud,
- relative humidity,
- vertical height of the cloud,
- mixing ratio of water vapour,
- mixing ratio of aerosol particles,
- mixing ratio of water drops,
- mixing ratio of ice particles,

It is necessary to divide the available data into training and test samples. Separation is performed randomly for each experiment in order to simulate the randomness of the nature of the occurrence of dangerous convective phenomena. Each parameter set is considered by the algorithm as a point in a 12-dimensional space.

## 3 Usage of k-Nearest Neighbour Algorithm

K-Nearest Neighbors algorithm (k-NN) is a lazy classifier. This means that in the process of learning it does not use new data, but only stores training data. It begins the classification only when new unmarked data appears.

Through the training sample, the algorithm remembered the coordinates of each set and the type (class) of the phenomenon. Each considered set of parameters from a test sample was matched by $k$ neighbors from the training sample, provided that the sum of the Euclidean distances to them was minimal. The value of $k$ was set manually, or the forecast accuracy was calculated for all values of $k$ from $k_0$ to $k_n$ in order to determine the value of $k_i$ at which the accuracy will be maximum. The Euclidean distance was expressed by the following formula:

$$D_E = \sqrt{\sum_{i}^{n} (x_i - y_i)^2},$$

where n is the number of parameters.

The considering set of parameters that characterized one of the types of a dangerous convective phenomenon was ranked as the class with the highest number of votes resulted from the neighbors voting. In the case of even number of classes, the value of k

was set to odd to improve the accuracy of the algorithm, because of an error appearing in the case of equality of votesfor even k.

As it was noted above, K-NN algorithm is cheap during training, but it can be expensive during classification, and, although it is usually very accurate, it can be subject to retraining.

Applying the k-NN method, we decided to improve it using the AdaBoost algorithm [10]. AdaBoost is sensitive to data noise and outliers, but it is less susceptible to retraining compared to other machine learning algorithms. AdaBoost was not used in the classic version, but was adapted to work with a strong classifier by using the operations of creating a committee of classifiers and weighted evaluation of a data set [19].

## 4 Usage of AdaBoost Algorithm

Unlike the lazy classifier, which begins the classification only when new unlabeled data appears, the active classifier creates a classification model in the learning process. When new data is entered, such a classifier adds data to the classification model. AdaBoost is an active classifier because it builds an ensemble classification.

This algorithm can be used in combination with several classification algorithms to improve their efficiency. The algorithm strengthens the classifiers, combining them into a "committee". AdaBoost is adaptive in the sense that each next classifier committee is built on objects that have been incorrectly classified by previous committees. AdaBoost is sensitive to data noise and outliers. However, it is less prone to retraining compared to other machine learning algorithms.

AdaBoost and weighted k-NN algorithms were implemented so that both of them complemented each other. AdaBoost was launched on the training set, creating weak classifiers for k-NN algorithm. Then the later ranin the feature space. Afterwards AdaBoost was used to improve classification accuracy and to prevent retraining by editing datasets provided by the weighted k-NN algorithm and thus improved the quality of training data. This joint algorithm (boosted k-NN) achieved significantly better classification accuracy than AdaBoost and kNN, used separately [20].

The detailed description of the joint algorithm is described below [21].

There is a sample $G$, which consists of n data sets (n = 676) that represent the output parameters of the model $X$ and the type of phenomenon $Y$.

Array X is the d-size vector of cloud characteristics $x_i$, $(i = 1, \ldots, d)$, $d$ is equal to 12.

Array$Y$ is the vector of available classes $y_i$, $(i = 1, \ldots, J)$. $J$ equals to 4 and represents the type of phenomenon (no phenomena, light rain, heavy rain, thunderstorm).

The sample $G$ was divided into 2 parts $G_t$ and $G_v$, $G_t$was the subset which was used for building classifiers, $G_v$ was the one for validating the obtained results. The division of the complete set of $G$ into $G_t$ and $G_v$ was carried out by randomly mixing the initial data set for each iteration of learning.

At the first stage, a set of weak classifiers was built.

The size of characteristic subset was chosen equal to $f$, where $f < d$. The more was the value of $f$, the stronger the classifier turned out. The best results were observed when $f$ was equal to$d/2$. We formed a set of randomly selected subsets of characteristics of size $n$. So each new data set $G_f$ contained $n$ subsets of the size $f$.

$G_f$ was used as a training sample for weak classifiers. We got $m$ training samples: $G_{f1}$, $G_{f2}$,.., $G_{fm}$, where $m$ is the number of weak classifiers (m = 12).

At the second stage fusion of classifier committee was performed.

1. Using the Gt sample, the accuracy of each weak classifier was calculated.
2. Classifiers were ranked according to the accuracy of the result, in descending order, using the same set of characteristics
3. The committee was formed from l most accurate classifiers. Best results were obtained when l was equal to 4.
4. To estimate the accuracy, the Brier score function (1) was used, showing the efficiency of the classifier forecast [22]:

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{1}$$

where $f_t$ is the forecast probability, $o_t$ is the outcome (0 or 1).

At the third stage the final decision was made on the type of convective phenomenon (forecast formation).

A weight, initially equal to 1/n, was assigned to each set of the training data. The most accurate weak classifier was the first to forecast. Next, the less accurate classifier worked, and it increased the weight of the data set under consideration in the case if its forecast differed from the one of the previous classifier. Otherwise it decreased the weight of the data set. Thus, each subsequent classifier spent more resources on processing data with the greater weight, and, accordingly, received more accurate results. Data that were classified uniquely required less cost. This saved computing resources, and the entire data set was processed faster.

A diagram describing the decision-making process by an ensemble of 3 classifiers is presented in Fig. 1.

## 5 Results

Three types of numerical studies were conducted.

The first concerned the determination of the accuracy of forecasting a specific type of convective phenomenon from four classes: no phenomena, light rain, heavy rain, thunderstorm. The calculation results showed that the boosted k-NN determined the type of the phenomenon with an accuracy that varied from 82% to 84% on average.

The accuracy of the k-NN algorithm without using Adaboost turned out to be significantly lower and varied from 42–61% on average.

The second type of numerical study concerned whether the phenomenon took place or not, without defining the type of it. The calculation results showed that the boosted k-NN algorithm determined the phenomenon occurrence with a high accuracy, achieving the value of 98%.

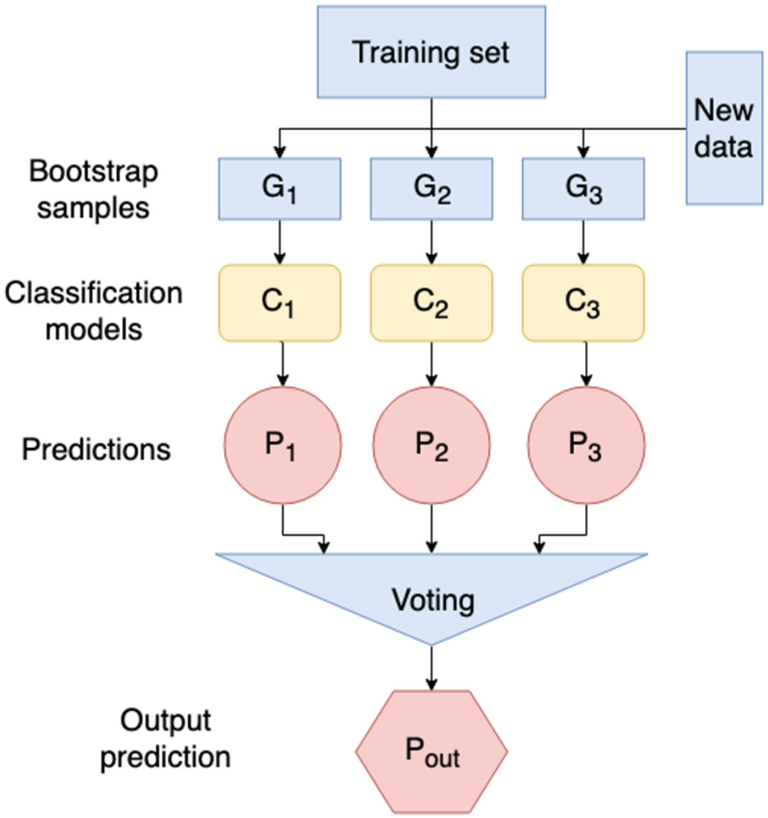


**Fig. 1.** Diagram describing the decision-making process by an ensemble of 3 classifiers

The accuracy of the k-NN algorithm without using Adaboost appeared to be less precise, but also high, achieving 96%.

The third research investigated the accuracy of the forecast of phenomenon occurrence, when the data set contained only pairs of classes: no phenomena and a light rain, no phenomena and a heavy rain, no phenomena and a thunderstorm.In this case boosted k-NN showed the highest accuracy in case of a thunderstorm (99%) and sufficiently lower accuracy for a light rain (86–89%) and for a heavy rain (88–92%).

The accuracy of the k-NN algorithm without using Adaboost was the same in case of a thunderstorm and lower for the other pairs 72–83% and 77–86% for a light and heavy rain correspondingly.

Figure 2 shows a comparative chart of the average accuracy of a forecast for the two variants of the k-NN algorithm.

All the calculations were provided using the open source library Scikit-learn [23]. It is written in the Python programming language and allows one to use a broad variety of machine learning algorithms. BSD (Berkeley Software Distribution) is a license distributer. Commercial use of the product is also allowed.

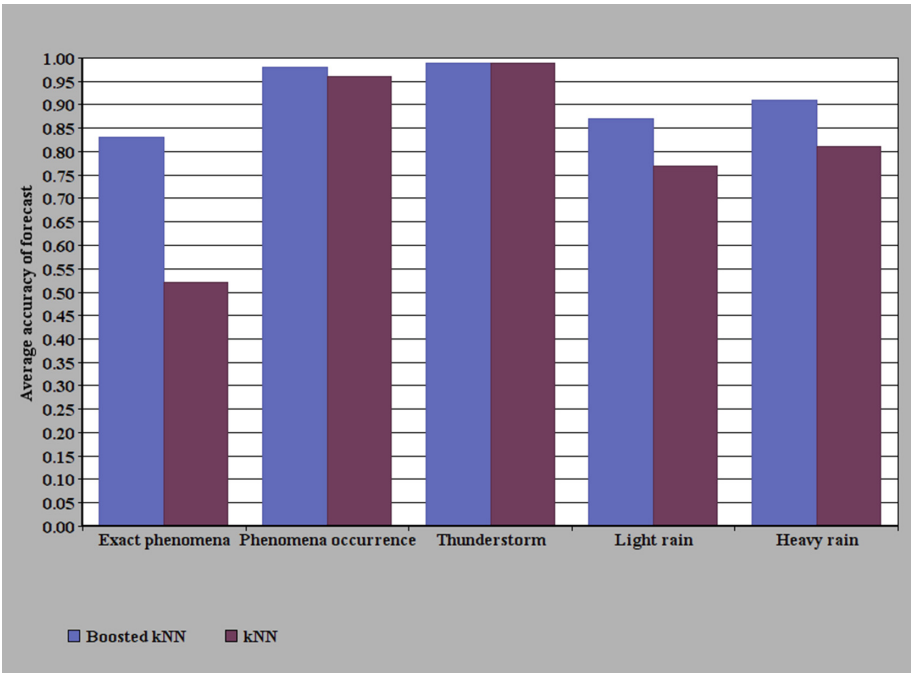


**Fig. 2.** Average accuracy of a forecast for the two variants of the k -NN algorithm

## 6 Conclusions

The boosted k-nearest neighbour algorithm was applied for processing the output data of the numerical cloud model for forecasting of convective phenomena (a thunderstorm, a heavy rain and a light rain). Boosting algorithm applied for k-nearest neighbour classifier was described in details.

The data set comprised 12 calculated dynamical and microphysical cloud characteristics and the parameter, describing the type of the phenomenon. Cloud characteristics were taken at the mature stage of evolution of the simulated cloud at a moment when the cloud achieved the maximum value of its upper boundary height and at the height corresponded to the maximum value of the mixing ratio of cloud drops.

Integrated information system [11–13] was applied for gathering of specific meteorological data used as the model input parameters. 480 radiosonde soundings were collected in the place and at the time when dangerous convective phenomena were observed, including 220 soundings for thunderstorms, 174 ones for heavy rains and 86 soundings for light rains. 196 soundings were collected in the cases when no dangerous convective phenomena were observed.

Three types of numerical studies were conducted with the aim to find out the cases when the processing of the cloud output data gave the highest values of forecast accuracy.

The first study tested the ability of the algorithm to distinguish the exact type of the phenomenon between four classes: no phenomena, a light rain, a heavy rain, and a thunderstorm. The second type of numerical study concerned whether the phenomenon took place or not, without defining the type of it. The third study investigated the accuracy of the forecast of phenomenon occurrence, when the data set contained only pairs of classes: no phenomena and a light rain, no phenomena and a heavy rain, no phenomena and a thunderstorm.

The research results showed that the highest accuracy of forecast (99%) was achieved in the third case when the pair no phenomena and a thunderstorm was considered. The lowest forecast accuracy (82 to 84%) was achieved in the first case study.

It can be noted in general that boosted k-NN algorithm was able to predict the very fact of convective phenomena occurrence with the high accuracy, but it was not so good in distinguishing the specific type of phenomena. It is planned to test the performance of the algorithm on a larger volume of input data obtained in different geographic regions.

Boosted k-NN algorithm provided better or equal results of forecast accuracy in all three cases in comparison with k-NN without boosting. The results obtained in the third case study for heavy and light rain are the only exception, where boosted k-NN algorithm provided significantly better classification accuracy.

In future we plan to include numerical cloud model and the block, processing the model output with the help of machine learning algorithms to the integrated information system [11–13] for providing the full cycle of forecasting of dangerous convective phenomena by joint system.

We plan that our improved and modified system should become a block of the Virtual private supercomputer [24, 25] and will be composed using the ideas presented in [26, 27]. The final aim is to enable users with the elaborated tool for individual forecast of the dangerous convective phenomena.

## References

1. Stankova, E.N., Balakshiy, A.V., Petrov, D.A., Shorov, A.V., Korkhov, V.V.: Using technologies of OLAP and machine learning for validation of the numerical models of convective clouds. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9788, pp. 463–472. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42111-7_36
2. Stankova, E.N., Grechko, I.A., Kachalkina, Y.N., Khvatkov, E.V.: Hybrid approach combining model-based method with the technology of machine learning for forecasting of dangerous weather phenomena. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10408, pp. 495–504. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62404-4_37
3. Stankova, E.N., Ismailova, E.T., Grechko, I.A.: Algorithm for processing the results of cloud convection simulation using the methods of machine learning. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10963, pp. 149–159. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95171-3_13
4. Stankova, E.N., Balakshiy, A.V., Petrov, D.A., Korkhov, V.V.: OLAP technology and machine learning as the tools for validation of the numerical models of convective clouds. Int. J. Bus. Intell. Data Min. **14**(1/2), 254–266 (2019)

5. Grover, A., Kapoor, A., Horvitz, E.: A deep hybrid model for weather forecasting research (2015). https://microsoft.com/en-us/um/people/horvitz/weather_hybrid_representation.pdf. Accessed 13 Aug 2016
6. Meteum technology. https://yandex.ru/pogoda/meteum. Accessed 17 Jan 2017. (in Russian)
7. The Weather Company Launches 'Deep Thunder' - the World's Most Advanced Hyper-Local Weather Forecasting Model for Businesses. https://www-03.ibm.com/press/us/en/pressrelease/49954.wss. Accessed 15 June 2016
8. How we perform a short-term forecast of precipitation. Lecture in Yandex. https://habr.com/ru/company/yandex/blog/328158/. Accessed 17 Jan 2017. (in Russian)
9. Fukunaga, K., Hostetler, L.: k-nearest-neighbor estimation. IEEE Trans. Inf. Theory **21**(3), 285–293 (1975)
10. Zhang, T.: Statistical behavior and consistency of classification methods based on convex risk minimization. Ann. Stat. **32**(1), 56–85 (2004)
11. Petrov, D.A., Stankova, E.N.: Use of consolidation technology for meteorological data processing. In: Murgante, B., et al. (eds.) ICCSA 2014. LNCS, vol. 8579, pp. 440–451. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09144-0_30
12. Petrov, D.A., Stankova, E.N.: Integrated information system for verification of the models of convective clouds. In: Gervasi, O., et al. (eds.) ICCSA 2015. LNCS, vol. 9158, pp. 321–330. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21410-8_25
13. Stankova, E.N., Petrov, D.A.: Complex information system for organization of the input data of models of convective clouds. Vestnik of Saint-Petersburg University. Series 10. Applied Mathematics. Computer Science. Control Processes. no 3, pp. 83–95 (2015). (in Russian)
14. Raba, N., Stankova, E.: Research of influence of compensating descending flow on cloud's life cycle by means of 1.5-dimensional model with 2 cylinders. In: Proceedings of MGO, vol. 559, pp. 192–209 (2009). (in Russian)
15. Raba, N., Stankova, E., Ampilova, N.: On investigation of parallelization effectiveness with the help of multi-core processors. Procedia Comput. Sci. **1**(1), 2757–2762 (2010)
16. Raba, N., Stankova, E.: On the possibilities of multi-core processor use for real-time forecast of dangerous convective phenomena. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6017, pp. 130–138. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12165-4_11
17. Raba, N.O., Stankova, E.N.: On the problem of numerical modeling of dangerous convective phenomena: possibilities of real-time forecast with the help of multi-core processors. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011. LNCS, vol. 6786, pp. 633–642. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21934-4_51
18. Raba, N.O., Stankova, E.N.: On the effectiveness of using the GPU for numerical solution of stochastic collection equation. In: Murgante, B., et al. (eds.) ICCSA 2013. LNCS, vol. 7975, pp. 248–258. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39640-3_18
19. La, L., Guo, Q., Yang, D., Cao, Q.: Multiclass boosting with adaptive group-based kNN and its application in text categorization. Math. Probl. Eng. **2012**, 24 (2012)
20. Yang, J.M., Yu, P.T., Kuo, B.C.: A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. IEEE Trans. Geosci. Remote Sens. **3**, 1279–1293 (2010)
21. Gul, A., et al.: Ensemble of a subset of kNN classifiers. Adv. Data Anal. Classif. **12**(4), 827 (2018)
22. Brier, G.W.: Verification of forecasts expressed in terms of probability (PDF). Mon. Weather Rev. **78**, 1–3 (1950)
23. Scikit-learn. Machine Learning in Python. http://scikit-learn.org/

24. Bogdanov, A., Degtyarev, A., Korkhov, V., Gaiduchok, V., Gankevich, I.: Virtual supercomputer as basis of scientific computing. In: Clary, T.S. (eds.) Horizons in Computer Science Research, vol. 11, pp. 159–198, Nova Science Publishers (2015). ISBN: 978-1-63482-499-6
25. Korkhov, V., Krefting, D., Kukla, T., Terstyanszky, G.Z., Caan, M., Olabarriaga, S.D.: Exploring workflow interoperability tools for neuroimaging data analysis. In: WORKS'11 - Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science, Co-located with SC 2011, pp. 87–96 (2011). https://doi.org/10.1145/2110497.2110508
26. Kulabukhova, N., Bogdanov, A., Degtyarev, A.: Problem-solving environment for beam dynamics analysis in particle accelerators. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications – ICCSA 2017. Lecture Notes in Computer Science, vol. 10408, pp. 473–482. Springer, Cham (2017)
27. Kulabukhova, N., Andrianov, S.N., Bogdanov, A., Degtyarev, A.: Simulation of space charge dynamics in high intensive beams on hybrid systems. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9786, pp. 284–295. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42085-1_22