

УДК 681.3

DOI: [10.26102/2310-6018/2019.27.4.040](https://doi.org/10.26102/2310-6018/2019.27.4.040)

МОДИФИКАЦИЯ НЕЙРОСЕТОВОЙ МОДЕЛИ RKELM С ДООБУЧЕНИЕМ

Ю.А. Асанов¹, С.Ю. Белецкая²

ФГБОУ ВО «Воронежский государственный технический университет»,
Воронеж, Российская Федерация

¹e-mail: su_bel@mail.ru

²e-mail: asanovjura@mail.ru

Резюме: Целью данной работы является разработка модели искусственной нейронной сети (ИНС), способной работать в динамически меняющихся условиях. Несмотря на большое количество исследований, и разработок в данной области, до сих пор отсутствуют модели, удовлетворяющие ограниченным условиям мобильных систем (прежде всего по производительности). В данной статье предлагается разработанная модификация модели с экстремальным обучением Хуанга (RKELM), отличающаяся от оригинального подхода процессом обучения (обучение на общих признаках, без увеличения матрицы весов и обучающей выборки, с последующим дообучением под конкретные условия). В качестве тестовой выборки данных использовался датасет из открытого репозитория машинного обучения UCI. Были поставлены вычислительные эксперименты, целью которых было выявление наиболее подходящей модели. Выбор производился из RKELM, SVM и ELM, критериями выбора являлись производительность и точность классификации. Наиболее подходящей оказалась модель с экстремальным обучением Хуанга, она и была использована в качестве основы разработанной модификации. Приведены результаты сравнения оригинальной и модифицированной модели. Предложенный подход превзошел конкурентные в скорости и производительности, незначительно уступая при этом лишь в точности классификации данных в изначальных условиях, но оказался значительно точнее в новых условиях, в которых модель не была обучена.

Ключевые слова: искусственная нейронная сеть, модификация RKELM, модель с дообучением.

Для цитирования: Асанов Ю.А., Белецкая С.Ю. Модификация нейросетевой модели rkelm с дообучением. *Моделирование, оптимизация и информационные технологии*. 2019;7(4).

Доступно по: https://moit.vivt.ru/wp-content/uploads/2019/11/AsanovBeletckaya_4_19_1.pdf DOI: 10.26102/2310-6018/2019.27.4.040

MODIFICATION OF NEURAL NETWORK MODEL RKELM WITH ADDITIONAL TRAINING

Y.A. Asanov, S.Y. Beletckaya

Voronezh State Technical University Voronezh, Russian Federation

Abstract: The aim of this work is developing of an artificial neural network model (ANN) capable of working in dynamically changing conditions. Despite a large number of research and development in this sphere, there are still no models that satisfy the limited resources of mobile systems (primarily – performance). This article proposes a developed modification of the Huang Extreme Learning Model, which differs from the original approach in the training process – training on common conditions, without increasing the weight matrix and the training sample, followed by further training for specific conditions. As a test sample of data, a dataset from the open source machine-learning repository UCI was used. Vast experiments were performed, the purpose of which was to identify the most suitable model, the choice was made from RKELM, SVM and ELM. The selection criteria for the model were performance and classification accuracy. The model with extreme training of Huang turned out to be the most suitable, it was used as the basis of the developed modification. The results of comparing the original and modified models are presented. The proposed approach surpassed the competition in speed

and performance, while only slightly inferior in accuracy of data classification in the initial conditions, but turned out to be much more accurate in the new conditions in which the model was not trained.

Keywords: artificial neural network, RKELM modification, model with an additional training.

For citation: Asanov Y.A., Beletckaya S.Y. Modification of neural network model RKELM with additional training. *Modeling, optimization and information technology*. 2019;7(4). Available by: https://moit.vivt.ru/wp-content/uploads/2019/11/AsanovBeletckaya_4_19_1.pdf DOI: 10.26102/2310-6018/2019.27.4.040 (In Russ.).

Введение

При разработке систем распознавания одной из наиболее сложных и актуальных задач является возможность применения системы в динамически меняющихся условиях. В ходе изучения текущих разработок в данной области было выявлено несколько подходов, которые можно условно разделить на два типа:

- 1) обобщающие все условия динамических сред;
- 2) подготовка к каждому условиям отдельных матриц весов с последующим переключением.

При первом подходе точность распознавания неизбежно падает. Для ее поддержания в приемлемом диапазоне авторы работы [1] предлагают увеличить тренировочную выборку, а также использовать модели ИНС, наиболее подходящие для решения текущей проблемы.

При втором подходе точность работы системы остается неизменной, так как одним из основных и важнейших ресурсов, задающим все характеристики ИНС (в том числе точность распознавания, процент ложных и ложноположительных результатов и т. д.), является матрица весов. В этом случае необходимо обучить множество отдельных, независимых систем, а также мастера, переключающего веса в соответствии с имеющимися условиями [2].

Проблемы обоих подходов очевидны – огромные тренировочные выборки, нелинейно растущая сложность обучения, большой объем весовых матриц, невозможность масштабирования системы.

В данной работе предлагается иной подход – обучение на общих признаках, без увеличения матрицы весов и обучающей выборки, с последующим дообучением под конкретные условия.

Особенности модели RKELM. Модель RKELM является развитием оригинального подхода ELM, отличительными особенностями которого являются следующие [1]:

- скрытые слои содержат узлы, а не нейроны. ELM вводит новое понятие – скрытый узел. Это вычислительный элемент, который выполняет ту же роль, что и нейрон, но в общем случае нейроном не является. Узел может быть любой нелинейной кусочно-непрерывной функцией (функцией Фурье, Гаусса, гиперболической функцией и т. д.), классическим искусственным нейроном или отдельной ИНС, сформированной из подмножества скрытых узлов [2, 3].

- отсутствие обратного распространения ошибки. Благодаря этому ELM получила серьезное преимущество в быстродействии (вплоть до тысячи кратного ускорения при определенных условиях [1]);

- отсутствие необходимости настройки параметров скрытых узлов. Все параметры задаются случайным образом (или наследуются от предка без изменения) и больше никогда не изменяются.

- в большинстве случаев выходные веса скрытых узлов обучаются за одну итерацию, что по существу сводится к обучению линейной модели. Это положительно сказывается на скорости обучения сетей с множеством скрытых слоев [4].

Несмотря на высокую производительность ELM, в некоторых случаях ее применение невозможно [3]. При попытке использовать ELM для обработки датасетов большого размера выяснилось, что:

- 1) матрица ядра $Q(X, X)$ может быть чрезмерно объемной что, как правило, приводит к нехватке памяти на компьютере;
- 2) время вычислений резко увеличивается с размером выборки, так что это становится нецелесообразным при применении в реальных условиях.

RKELM была создана для преодоления этих ограничений. В рамках работы RKELM производится выбор небольшого случайного подмножества $\check{X} = \{x_i\}_{i=1}^{\check{n}}$ из исходного датасета $X = \{x_i\}_{i=1}^n$, где $\check{n} \ll n$, после чего вместо $Q(X, X)$ используется $Q(X, \check{X})$ – редуцированная матрица, что приведет к сокращению необходимого объема памяти и уменьшит время вычислений. Выходная функция RKELM имеет следующий вид [5, 6]:

$$f(x) = \begin{pmatrix} Q(x, x_1) \\ \vdots \\ Q(x, x_{\check{n}}) \end{pmatrix}^T \left(\frac{1}{C} + Q(X, \check{X})^T Q(X, \check{X}) \right)^{-1} Q(X, \check{X})^T T$$

где T – целевая матрица обучения, C – стабилизирующий коэффициент.

Алгоритм RKELM можно обобщить следующим образом [5]:

- 1) Выбрать случайное подмножество $\check{X} = \{x_i\}_{i=1}^{\check{n}}$; Обычно $\check{n} \ll n$.
- 2) Построить прямоугольную матрицу ядра $Q(X, \check{X})$;
- 3) Получить прогноз для новых данных x .

Модификация модели RKELM с дообучением. Модель RKELM обладает выдающимися результатами в задачах обобщения, превосходящими даже результаты SVM (Support Vector Machine) [3, 4]. Но при подробном изучении RKELM становится очевидно, что применение этой модели на динамически меняющихся источниках данных нецелесообразно. Переобучение RKELM хоть и проходит быстрее, чем у ELM, но все же недопустимо долго, соответственно не может проходить в режиме реального времени. Предложенный подход, основанный на модели с экстремальным обучением Хуанга (RKELM Хуанга) [1], свободен от данного недостатка. Схема алгоритма представлена на Рисунке 1.

Предложенный алгоритм сводится к двум этапам: первичное построение модели классификации и ее дообучение.

На первом шаге происходит основное построение модели классификации, а также ведется распознавание. В процессе формирования модели классификации, исходные скалярные данные импортируются и затем векторизуются для того чтобы исключить обучение нейронной сети по ориентационному признаку. Получившаяся на данном этапе обучающая выборка передается в RKELM-классификатор, параметры которого выставлены таким образом, чтобы увеличить скорость обучения, но в то же время существенно не уменьшить производительность.

После первичного обучения ИНС начинается процесс дообучения. При получении результатов работы на исходных данных формируется новая обучающая выборка. Данные, полученные в результате работы классификатора, имеют некую оценку достоверности. Для последующего процесса дообучения используются только те данные, достоверность которых выше заранее выставленного (эмпирически

подобранного) порога t . Матрица весов классификатора обновляется с помощью полученной обучающей выборки, после чего начинается новая итерация обучения.

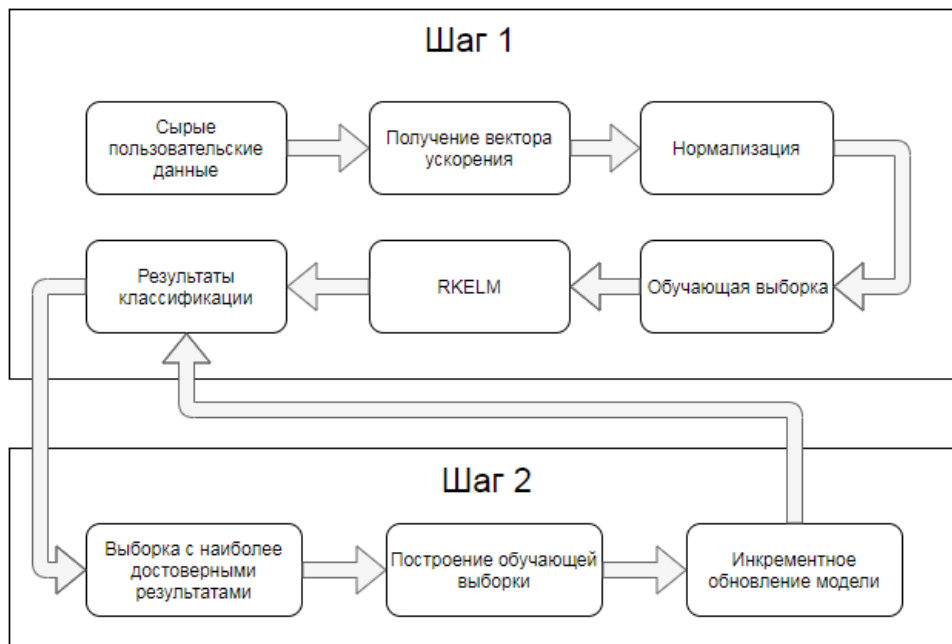


Рисунок 1 – Схема алгоритма классификации с дообучением
Figure 1 – Diagram of the classification algorithm with retraining

Адаптация модели к новым источникам данных (пользователям) происходит по следующему сценарию:

1. В процессе работы модели формируются новые обучающие выборки, состоящие из результатов классификации с достоверностью выше порогового значения t .

2. При достижении достаточного числа обучающих выборок начинается процесс постепенного обновления (количество выборок, при котором целесообразно начало дообучения, подбирается исходя из конкретных требований к системе).

Исходя из вышеописанных правил, алгоритм дообучения может быть формализован следующим образом [5, 7]:

Пусть α^0 – весовая матрица, полученная для выборки X^0 , тогда X' – обучающий набор данных, взятый случайным образом из подготовленного в процессе работы системы пула, H^0 – временная матрица, необходимая для подсчета весов, которая вычисляется следующим образом:

$$H^0 = (\partial^{-1} + Q_0^K Q_0)^{-1}.$$

В таком случае, обучающая выборка, которая будет использоваться на текущем этапе дообучения [8] - $\{X^1 = \{x_i\}_{i=1}^{n_1}, K^1 = \{k_i\}_{i=1}^{n_1}\}$,

где ∂ – предустановленная положительная константа;

Q_0^K – окно сверточной сети.

1) Вычислить матрицу окна $Q^1 = \Theta(\alpha^1, \alpha')$;

2) Рассчитать $H^1 = H^0 - H^0 Q_1^K (1 + Q_1 H^0 Q_1^K)^{-1} Q_1 H^0$;

3) Рассчитать веса $\alpha^1 = \alpha^0 + H^1 Q_1^K (K^1 - Q_1 \alpha^0)$.

В данной работе используется набор данных, собранный с 30 пользователей в возрасте от 19 до 48 лет, который может быть загружен из репозитория машинного обучения UCI [6]. Каждый человек выполнял шесть действий (ходить, подниматься по

лестнице, спускаться по лестнице, сидеть, стоять, лежать), во время этих упражнений каждый пользователь имел при себе смартфон, который регистрировал ускорение и угловую скорость по 3 осям. В наших экспериментах данные случайным образом были разделены на два набора, где 80 % использовались для обучения, оставшиеся 20 % для тестирования. В Таблице 1 показано количество образцов, полученных для каждого вида деятельности.

Таблица 1 – Тестовый набор данных
Table 1 – Test Dataset

Упражнение	Число выборок
Ходить	1722
Подниматься по лестнице	1544
Спускаться по лестнице	1406
Сидеть	1777
Стоять	1906
Лежать	1944
Итого	10299

Для данной работы был использован именно RKELM из-за его достаточно высокой точности и превосходной производительности, что доказывается нижеописанными экспериментами.

Было проведено сравнение производительности RKELM, ELM и SVM по распознаванию физической активности пользователей.

При проведении экспериментов были использованы следующие параметры: RKELM - $\sigma = 1024$, $\partial = 2^{30}$, $\hat{n} = 500$ (число скрытых узлов); ELM - $\gamma = 1$, $\hat{n} = 1000$; SVM - $\partial = 2^{15}$.

В Таблице 2 приведены: время обучения (в секундах), время тестирования (в секундах) и точность тестирования.

Таблица 2 – Сравнение производительности классификаторов
Table 2 – Comparison of the performance of classifiers

	RKELM	ELM	SVM
Время обучения (с.)	0,64	2,09	5,1
Время работы (с.)	0,12	0,18	2,83
Точность (%)	98,49	99,05	98,77

Как видно из Таблицы 2, точность RKELM – 98,49 %, что можно назвать конкурентноспособной точностью (хотя и не самой высокой). Кроме того, время обучения и работы RKELM намного меньше, чем у SVM и ELM. Среднее время обучения RKELM составляет 0,64 с, в то время как ELM и SVM требуют 2,09 и 5,1 секунду соответственно. Из чего можно сделать вывод, что RKELM намного быстрее, чем SVM и ELM.

Чтобы оценить ухудшение производительности распознавания активности при смене пользователя, проводится эксперимент по распознаванию активностей людей, на которых сеть не была натренирована, с отключенным функционалом адаптивования. Для каждого человека, был обучен отдельный классификатор, и затем проверен на всех

людях. Для этого эксперимента была случайным образом выбрана группа из 10 пользователей.

Таблицы 3, 4 и 5 отображают полученные результаты для классификаторов RKELM, ELM и SVM соответственно (округлены до целых значений).

Таблица 3 – Точность классификации RKELM

Table 3 – RKELM classification accuracy

	п1	п2	п3	п4	п5	п6	п7	п8	п9	п10
п1	100	78	85	77	78	75	74	86	59	70
п2	91	93	87	73	67	78	75	80	60	60
п3	88	90	100	67	65	76	87	70	50	50
п4	76	80	93	98	87	84	65	67	43	56
п5	57	81	81	88	100	91	64	64	57	72
п6	91	79	81	86	82	97	78	74	37	53
п7	87	83	87	75	82	76	100	78	77	60
п8	92	84	70	67	98	64	73	100	75	65
п9	75	77	73	72	68	68	72	67	87	87
п10	74	80	82	78	78	72	56	77	73	100

Таблица 4 – Точность классификации ELM

Table 4 – ELM classification accuracy

	п1	п2	п3	п4	п5	п6	п7	п8	п9	п10
п1	100	62	67	68	63	56	53	63	40	64
п2	56	98	63	64	56	52	61	42	55	45
п3	43	65	100	59	40	46	55	46	48	32
п4	34	48	64	98	43	54	49	43	39	36
п5	50	63	68	69	100	59	46	47	43	41
п6	47	47	63	62	74	100	40	39	32	43
п7	33	57	52	36	33	37	100	35	51	25
п8	70	27	57	58	61	54	50	100	43	44
п9	23	53	30	41	35	24	47	30	78	42
п10	44	60	56	57	68	46	44	49	55	100

Таблица 5 – Точность классификации SVM

Table 5 – SVM classification accuracy

	п1	п2	п3	п4	п5	п6	п7	п8	п9	п10
п1	100	73	87	87	76	72	80	81	73	84
п2	87	90	91	78	76	80	79	79	80	77
п3	92	91	100	58	74	75	92	92	68	63
п4	77	90	92	100	87	79	76	76	72	71
п5	77	82	81	87	100	92	81	81	66	79
п6	91	81	85	82	86	100	82	82	57	83
п7	92	86	86	71	78	74	100	72	81	73
п8	93	90	68	64	69	67	70	100	72	85
п9	83	79	84	81	72	69	83	77	100	82
п10	85	85	79	77	89	87	79	76	80	97

Значение элемента i -й строки и j -го столбца представляет точность классификации, произведенной классификатором, обученным на i -м пользователе и протестированном на j -м пользователе.

Из таблиц можно сделать вывод о том, что RKELM и ELM одинаково эффективны при применении на известных пользователях. Но точность ELM значительно снижается при применении на новых пользователях. SVM имеет незначительное преимущество перед RKELM при работе с известными пользователями, и схожую точность при работе с неизвестными пользователями.

Далее рассмотрим полученные результаты при использовании адаптации к новым пользователям.

В качестве примера были использованы следующие данные: первоначальная обучающая выборка была сформирована из датасета, который был сформирован согласно сценарию – были случайно выбраны пользователи U_1 и U_2 . Соответственно, показания их акселерометров - D_1 и D_2 . Так же, случайным образом был выбран неизвестный системе пользователь U_3 с набором данных D_3 . Данные пользователей были случайным образом разделены в соотношении 80% (D_{1L}, D_{2L}, D_{3L}) и 20% (D_{1U}, D_{2U}, D_{3U}). Тогда:

$U_1U_{2_{обуч}} = D_{1L} + D_{2L}$ – выборка, которая используется для обучения модели RKELM в начальный момент. D_{3L} – датасет неизвестного пользователя, будет применяться в процессе дообучения. $U_1U_{2_{тест}} = D_{1U} + D_{2U}$ – тестовый датасет, который будет показателем эффективности применения модели на известных пользователях. D_{3U} – тестовый датасет, который будет показателем эффективности применения модели на неизвестном пользователе. Для тестового прогона был выбран порог $t = 0.75$. D_{AL} – датасет дообучения, сформирован из данных, полученных в процессе работы модели на неизвестных данных (в нашем случае является подмножеством множества D_{3L}).

Таблица 6 – Результаты работы ИНС для известных пользователей

Table 6 – the results of the ANN for well-known users

	До дообучения	После дообучения
Обучающий датасет	$U_1U_{2_{обуч}}$	$U_1U_{2_{обуч}} + D_{AL}$
Тестовый датасет	$U_1U_{2_{тест}}$	$U_1U_{2_{тест}}$
Точность (%)	96,95	97,71

Таблица 7 – Результаты работы ИНС для неизвестного пользователя

Table 7 – ANN results for an unknown user

	До дообучения	После дообучения
Обучающий датасет	$U_1U_{2_{обуч}}$	$U_1U_{2_{обуч}} + D_{AL}$
Тестовый датасет	D_{3U}	D_{3U}
Точность (%)	88,41	92,75

В Таблице 6 отображены результаты работы модели на известных пользователях, в первой колонке – до дообучения, во второй – после. Значительного изменения точности не наблюдается (увеличение точности $< 1\%$). Таблица 7 отображает точность классификации данных неизвестного системе пользователя. Прирост точности в этом случае уже можно назвать существенным ($\sim 4,3\%$).

Заключение

Исходя из вышесказанного, становится очевидно, что применение описанного в данной работе алгоритма способно принести значительный прирост в точности распознавания данных, полученных от неизвестных системе источников данных. Кроме того, благодаря малоразмерной матрице весов и ее «дешевому» обновлению, данный алгоритм применим на устройствах с ограниченным запасом ресурсов – памяти, вычислительной мощности, емкости устройства питания.

ЛИТЕРАТУРА

1. Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. *Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine*. In *Ambient assisted living and home care*-Springer. 2012:216–223.
2. Львович Я.Е., Белецкая С.Ю. Повышение эффективности процедур параметрического синтеза сложных систем на основе трансформации оптимизационных задач. *Информационные технологии*. 2002;10:31-35.
3. Huang, G.B., Zhu, Q.Y., & Siew, C.K. *Extreme learning machine: theory and applications*. *Neurocomputing*. 2006:489–501.
4. Lee, Y.J., & Huang, S.Y. *Reduced support vector machines: a statistical theory*. *Neural Networks, IEEE Transactions on*. 2007:1–13.
5. Deng, W., Zheng, Q., & Zhang, K. *Reduced Kernel Extreme Learning Machine*. In *Proceedings of the 8th international conference on computer recognition systems CORES*. 2013:63–69.
6. Blake, C.L., & Merz, C.J. *UCI Repository of machine learning databases* <http://www.ics.uci.edu/mllearn/MLRepository.html> - Irvine, CA: University of California, Department of Information and Computer Science. 1998:55–78.
7. Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. *Cascaded pyramid network for multi-person pose estimation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018:7103–7112.
8. Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. *Searching for efficient multi-scale architectures for dense image prediction*. In *Advances in Neural Information Processing Systems*. 2018:8699–8710.

REFERENCES

1. Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. *Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine*. In *Ambient assisted living and home care*-Springer. 2012:216–223.
2. Lvovich Ya.E., Beletskaya S.Yu. Increasing the efficiency of parametric synthesis procedures for complex systems based on the transformation of optimization problems. *Information technologies*. 2002;10:31–35.
3. Huang, G.B., Zhu, Q.Y., & Siew, C.K. *Extreme learning machine: theory and applications*. *Neurocomputing*. 2006:489–501.
4. Lee, Y.J., & Huang, S.Y. *Reduced support vector machines: a statistical theory*. *Neural Networks, IEEE Transactions on*. 2007:1–13.
5. Deng, W., Zheng, Q., & Zhang, K. *Reduced Kernel Extreme Learning Machine*. In *Proceedings of the 8th international conference on computer recognition systems CORES*. 2013:63–69.

6. Blake, C.L., & Merz, C.J. *UCI Repository of machine learning databases* <http://www.ics.uci.edu/mllearn/MLRepository.html> - Irvine, CA: University of California, - Irvine, CA: University of California, Department of Information and Computer Science. 1998:55–78.
7. Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. *Cascaded pyramid network for multi-person pose estimation*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018:7103–7112.
8. Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. *Searching for efficient multi-scale architectures for dense image prediction*. In *Advances in Neural Information Processing Systems*. 2018:8699–8710.

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHORS

Асанов Юрий Анатольевич, аспирант, кафедра систем автоматизированного проектирования и информационных систем, Воронежский государственный технический университет, Воронеж, Российская Федерация.

Yury A. Asanov, P.G., Department of Automated Design and Information Systems, Voronezh State Technical University, Voronezh, Russian Federation.

Белецкая Светлана Юрьевна, профессор, доктор технических наук, кафедра систем автоматизированного проектирования и информационных систем, Воронежский государственный технический университет, Воронеж, Российская Федерация.

Svetlana Y. Beletskaya, prof., PhD, Department of Automated Design and Information Systems, Voronezh State Technical University, Voronezh, Russian Federation.