

Седых, И. А. Нейро-окрестностные модели как новый класс иерархических динамических окрестностных моделей / И. А. Седых, В. А. Истомин // Прикладная математика и вопросы управления. – 2024. – № 2. – С. 34–48. DOI 10.15593/2499-9873/2024.2.03

Библиографическое описание согласно ГОСТ Р 7.0.100–2018

Седых, И. А. Нейро-окрестностные модели как новый класс иерархических динамических окрестностных моделей / И. А. Седых, В. А. Истомин. – Текст : непосредственный. – DOI 10.15593/2499-9873/2024.2.03 // Прикладная математика и вопросы управления / Applied Mathematics and Control Sciences. – 2024. – № 2. – С. 34–48.



**пермский
политех** ПРИКЛАДНАЯ МАТЕМАТИКА
И ВОПРОСЫ УПРАВЛЕНИЯ
№ 2, 2024

<https://ered.pstu.ru/index.php/amcs>



Научная статья

DOI: 10.15593/2499-9873/2024.2.03

УДК 519.711, 004.852



Нейро-окрестностные модели как новый класс иерархических динамических окрестностных моделей

И.А. Седых, В.А. Истомин

Липецкий государственный технический университет, Липецк, Российская Федерация

О СТАТЬЕ

Получена: 22 мая 2024
Одобрена: 14 июля 2024
Принята к публикации:
12 августа 2024

Финансирование

Исследование не имело спонсорской поддержки.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Вклад авторов

равноценен.

Ключевые слова:

имитационные модели, нейронные сети, окрестностные модели, иерархические динамические нейро-окрестностные модели, Python.

АННОТАЦИЯ

Представлены основные особенности моделирования сложных распределенных процессов, отражена актуальность исследования и важность моделирования таких процессов. Рассматривается развитие окрестностного подхода, труды отечественных и зарубежных авторов, внесших значительный вклад в развитие математического моделирования сложных динамических систем. Приведены виды окрестностных моделей и отражено положение нового направления иерархических динамических нейро-окрестностных моделей в классе окрестностных моделей. Представлены преимущества развития данного подхода, а именно улучшение интерпретируемости модели при одновременном обеспечении достаточной точности построения и общая схема узла исследуемой модели по переходам и выходам. Теоретико-множественный способ описывает модель в виде множеств узлов и иерархических окрестностных связей между ними. Матричный способ позволяет представить модель в виде матриц смежности для переходов и выходов по состояниям и по внешним воздействиям соответственно. Приведено подробное описание иерархических динамических нейро-окрестностных моделей и нейронных сетей в узлах. Описан алгоритм идентификации разработанного подхода, показана схема алгоритма идентификации. Приведен пример построения иерархической динамической нейро-окрестностной модели прогноза общего энергопотребления бытовой техники в доме с учетом отопления и погодных условий в реализованной программе Python с автоматическим подбором оптимальных параметров модели. Приведено описание исходных данных, взятых с сайта Kaggle. Проведена подготовка данных, на основе которых выполнено обучение и тестирование полученной модели. Показана схема иерархической динамической нейро-окрестностной модели прогнозируемого процесса. Сделаны выводы по проделанному исследованию.

© Седых Ирина Александровна – доктор технических наук, профессор кафедры автоматизированных систем управления, e-mail: sedykh-irina@yandex.ru, ORCID: 0000-0003-0012-8103.

Истомин Владимир Александрович – аспирант кафедры автоматизированных систем управления, e-mail: istomin96@mail.ru, ORCID: 0000-0003-3010-1470.



Эта статья доступна в соответствии с условиями лицензии Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0)

Perm Polytech Style: Sedykh I.A., Istomin V.A. Neuro-neighborhood models as a new class of hierarchical dynamic neighborhood models. *Applied Mathematics and Control Sciences*. 2024, no. 2, pp. 34–48. DOI: 10.15593/2499-9873/2024.2.03

MDPI and ACS Style: Sedykh, I.A.; Istomin, V.A. Neuro-neighborhood models as a new class of hierarchical dynamic neighborhood models. *Appl. Math. Control Sci.* **2024**, *2*, 34–48. <https://doi.org/10.15593/2499-9873/2024.2.03>

Chicago/Turabian Style: Sedykh, Irina A., and Vladimir A. Istomin. 2024. “Neuro-neighborhood models as a new class of hierarchical dynamic neighborhood models”. *Appl. Math. Control Sci.* no. 2: 34–48. <https://doi.org/10.15593/2499-9873/2024.2.03>



APPLIED MATHEMATICS
AND CONTROL SCIENCES

№ 2, 2024

<https://ered.pstu.ru/index.php/amcs>



Article

DOI: 10.15593/2499-9873/2024.2.03

UDK 519.711, 004.852



Neuro-neighborhood models as a new class of hierarchical dynamic neighborhood models

I.A. Sedykh, V.A. Istomin

Lipetsk State Technical University, Lipetsk, Russian Federation

ARTICLE INFO

Received: 22 May 2024
Approved: 14 July 2024
Accepted for publication:
12 August 2024

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

equivalent.

Keywords:

simulation models, neural networks, neighborhood models, hierarchical dynamic neuro-neighborhood models, Python.

ABSTRACT

The article presents the main features of modeling complex distributed processes, reflects the relevance of the research and the importance of modeling such processes. The development of the neighborhood approach and the works of domestic and foreign authors who have made a significant contribution to the development of mathematical modeling of complex dynamic systems are considered. The types of neighborhood models are presented and the position of the new direction of hierarchical dynamic neuro-neighborhood models in the class of neighborhood models is reflected. The advantages of developing this approach are presented, namely, improving the interpretability of the model while simultaneously ensuring sufficient accuracy with generalization ability and resistance to noise. The main stages of construction are highlighted and the areas of application of hierarchical dynamic neuro-neighborhood models are presented. There are 3 ways of representing their structure: graphical, set-theoretic and matrix. The graphical presentation method is based on graphs divided into two layers, which describe the connections between nodes by transitions and outputs, respectively. The diagrams of layers and the general diagram of the node of the model under study are shown according to transitions and outputs. The set-theoretic method describes the model in the form of sets of nodes and hierarchical neighborhood connections between them. The matrix method allows you to present the model in the form of adjacency matrices for transitions and outputs by state and by external influences, respectively. A detailed description of hierarchical dynamic neuro-neighborhood models and neural networks in nodes is provided. The identification algorithm of the developed approach is described, and a diagram of the identification algorithm is shown. An example is given of constructing a hierarchical dynamic neuro-neighborhood model for predicting the total energy consumption of household appliances in a house, taking into account heating and weather conditions in the implemented Python program with automatic selection of optimal model parameters. A description of the source data taken from the Kaggle website is provided. The data was prepared, on the basis of which the resulting model was trained and tested. A diagram of a hierarchical dynamic neuro-neighborhood model of the predicted process is shown. Conclusions are drawn from the research done.

© Irina A. Sedykh – Doctor of Technical Sciences, Professor of the Department of Automated Control Systems, e-mail: sedykh-irina@yandex.ru, ORCID: 0000-0003-0012-8103.

Vladimir A. Istomin – Ph. D. Student of the Department of Automated Control Systems, e-mail: istomin96@mail.ru, ORCID: 0000-0003-3010-1470.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0)

Введение

Сложные распределенные процессы представляют собой фундаментальную область исследований, охватывающую различные направления науки и техники. В современном мире, где информационные технологии, биоинженерия, финансовые рынки и промышленные системы становятся все более сложными, понимание и моделирование таких распределенных процессов являются неотъемлемой задачей.

Актуальность темы проявляется во взаимосвязи сложных систем, где множество взаимодействующих элементов формируют динамические структуры. Например, в области информационных технологий распределенные вычисления, облачные сервисы и сетевые протоколы требуют глубокого понимания сложных распределенных процессов для оптимизации производительности и безопасности.

Примерами могут служить распределенные базы данных, где данные хранятся и обрабатываются на различных узлах сети, или глобальные финансовые рынки, где транзакции моментально распространяются по всему миру. Распределенные процессы также встречаются в биологии, например, в моделировании сложных взаимодействий в биохимических системах.

Математическое моделирование сложных распределенных процессов играет ключевую роль в предсказании и оптимизации их поведения. Понимание структуры и динамики таких процессов позволяет разрабатывать эффективные стратегии управления и принятия решений. Дальнейшие исследования в этой области способствуют не только улучшению технологий, но и открывают новые горизонты для научных открытий и инноваций [1–7].

Множество отечественных и зарубежных ученых внесли значительный вклад в развитие математического моделирования сложных динамических систем. Окрестностный подход, предложенный в 90-е гг. XX в. учеными С.Л. Блюминым и А.М. Шмыриным, представляет собой развитие теории математического моделирования дискретных детерминированных распределенных систем [8]. Он отличается от других подходов тем, что позволяет гибко определять связи между элементами системы с использованием окрестностей и представлять структуру системы в виде графа с различными видами дуг. В рамках этого подхода исследованы различные виды статических окрестностных моделей, такие как линейные, билинейные, полилинейные, четкие и нечеткие, с фиксированными окрестностями. Также были разработаны методы параметрической идентификации и смешанного управления для этих моделей. Следует отметить, что схожая тематика, связанная с «окрестностями» и «окрестностными связями», рассматривается в работах Y. Shang [9]. В этих работах используются агенты, перемещающиеся по окрестностям и взаимодействующие друг с другом согласно определенным правилам. Кроме того, введено понятие динамических окрестностных моделей в работах А.А. Томилина [10]. Однако в этих моделях не рассматриваются вопросы нечеткости как в отношении связей, так и в отношении передаваемых сигналов.

Затем были разработаны классы динамических окрестностных моделей на основе сетей Петри, как четкие, так и нечеткие, которые отличаются от предыдущих моделей изменением окрестностных связей в процессе функционирования. Нечеткие модели объединяют нечеткость по значениям и окрестностным связям между узлами, что позволяет моделировать недетерминированные параллельные процессы, которые характерны для сложных распределенных систем. Также были разработаны методы идентификации и решения задач, связанных с достижимостью состояний таких систем [11].

Однако стоит отметить, что четкие окрестностные модели сетей Петри ограничены в своей способности моделирования, так как они являются линейными и предполагают, что

состояния системы являются целыми числами и не могут быть отрицательными. Это сужает возможности моделирования более широкого класса систем. Кроме того, как в четких, так и в нечетких окрестностных моделях сетей Петри отсутствует учет внешнего воздействия на процесс. Все ранее описанные окрестностные модели являются одноуровневыми.

Позже были разработаны классы окрестностных моделей с иерархическими окрестностными связями между элементами системы. Использование математического аппарата динамического окрестностного подхода основывается на учете уже известных структурных связей между отдельными компонентами системы, при этом нет необходимости в задании их функционального взаимодействия. Это позволяет говорить о том, что такие модели и методы являются универсальными.

Закономерным этапом развития иерархического динамического окрестностного подхода, обладающего рядом преимуществ, является объединение его возможностей с другими моделями, например нейронными сетями [12–14], улучшая интерпретируемость модели и одновременно обеспечивая достаточную точность. Такие модели обладают высокой обобщающей способностью и устойчивостью к шуму. Они идеально подходят для решения сложных задач, где требуется комбинированный подход. Однако создание таких моделей может потребовать специальной технической подготовки для их интеграции и совместной работы. Обязательным этапом для построения модели является анализ прогнозируемого процесса или системы. На основе проведенного анализа выполняется построение структуры: выделение узлов и связей между ними, затем задание внешних воздействий, состояний и выходов модели.

Идея иерархических динамических нейро-окрестностных моделей (ИДНО) [15–18] состоит в представлении сложных систем и их подпроцессов с помощью объединения окрестностной структуры с различными видами нейронных сетей (рис. 1). Узлы окрестностной модели являются иерархическими с вложенными в них нейронными сетями.

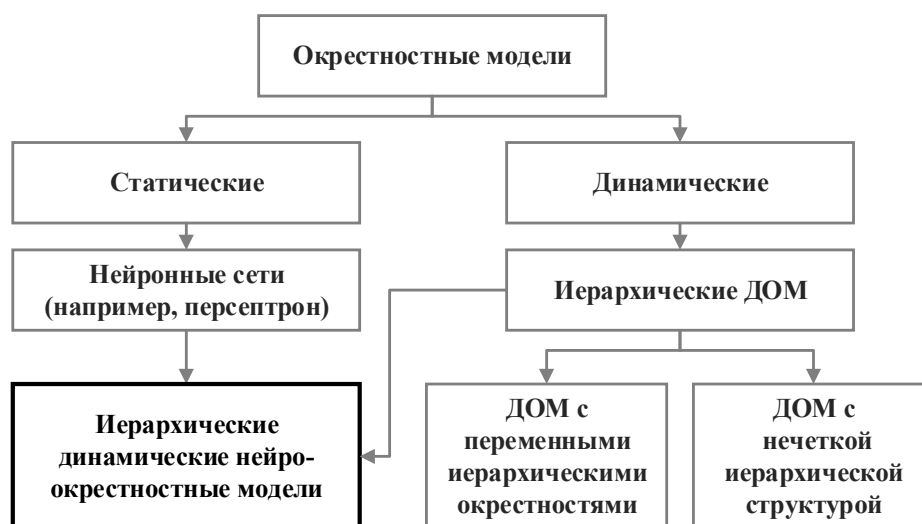


Рис. 1. Обобщенная схема связей классов окрестностных моделей

На рис. 1 изображена обобщенная схема, отображающая статические и динамические окрестностные модели, которые включают в себя соответственно нейронные сети, иерархические динамические окрестностные модели с переменными иерархическими окрестностями моделями и с нечеткой иерархической структурой. Класс иерархических динамических нейро-окрестностных моделей является направлением на стыке нейронных сетей и иерархических динамических окрестностных моделей.

Использование ИДНО моделей возможно в таких сферах, как металлургическое и цементное производства, транспортные модели, экономические и социальные процессы и т.д.

В рамках данной статьи будет приведено описание ИДНО моделей.

Иерархические динамические нейро-окрестностные модели

ИДНО модели NS_{IER}^{NN} имеют три способа представления их структуры: графический, теоретико-множественный и матричный.

Графический способ

Структура ИДНО модели состоит из двух слоев, каждый из которых описывает связи между узлами модели по переходам и по выходам модели соответственно. Структуру каждого слоя можно описать с помощью графов. На рис. 2 представлен пример графа структуры модели по переходам, а на рис. 3 – по выходам.

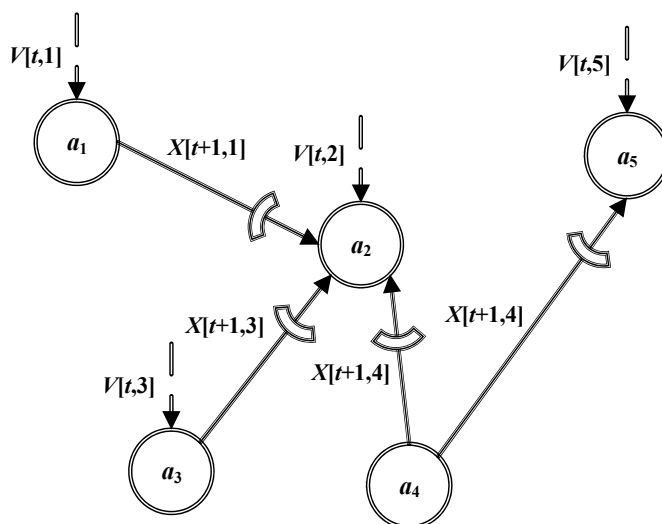


Рис. 2. Пример графа структуры ИДНО модели по переходам

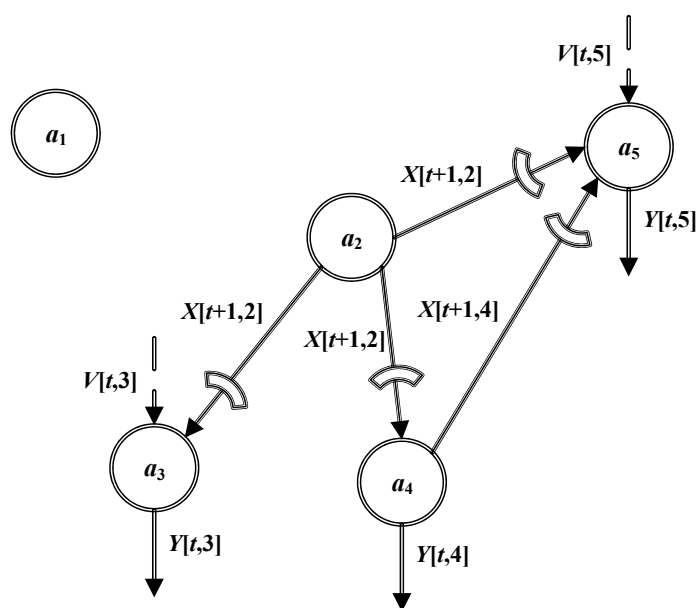


Рис. 3. Пример графа структуры ИДНО модели по выходам

На рис. 2, 3 a_i – узел, в котором находятся нейронные сети для пересчета состояний и для пересчета выходов соответственно, где $i = 1, \dots, n$, n – количество узлов; символ ζ – задержка на один такт функционирования модели; $V[t, i]$ – внешнее воздействие на узел a_i , где t – это момент времени; $X[t, i]$ – состояния узла a_i , $Y[t, i]$ – это выход узла a_i .

Окрестностью $O_x^{(x)}[a_i]$ узла a_i для переходов по состояниям назовем множество узлов, состояние которых влияет на этот узел. Аналогично определяется окрестность для переходов по внешним воздействиям $O_v^{(x)}[a_i]$, окрестности для выходов по состояниям $O_x^{(y)}[a_i]$ и по внешним воздействиям $O_v^{(y)}[a_i]$.

Введем обозначения: $X_i[t] = (X_i^{(x)}[t], X_i^{(y)}[t])$; $V_i[t] = (V_i^{(x)}[t], V_i^{(y)}[t])$;

$$X_i^{(x)}[t] = (X^{(x)T}[t, j_1], \dots, X^{(x)T}[t, j_r])^T;$$

$$V_i^{(x)}[t] = (V^{(x)T}[t, k_1], \dots, V^{(x)T}[t, k_h])^T;$$

$$X_i^{(y)}[t] = (X^{(y)T}[t, p_1], \dots, X^{(y)T}[t, p_e])^T;$$

$$V_i^{(y)}[t] = (V^{(y)T}[t, u_1], \dots, V^{(y)T}[t, u_d])^T,$$

где для всех $s1 = 1, \dots, r$ узел $a_{j_{s1}} \in O_x^{(x)}[a_i]$; для всех $s2 = 1, \dots, h$ узел $a_{k_{s2}} \in O_v^{(x)}[a_i]$; для всех $s3 = 1, \dots, e$ узел $a_{p_{s3}} \in O_x^{(y)}[a_i]$; для всех $s4 = 1, \dots, d$ узел $a_{u_{s4}} \in O_v^{(y)}[a_i]$.

Схема нейронной сети в узле a_i представлена на рис. 4.

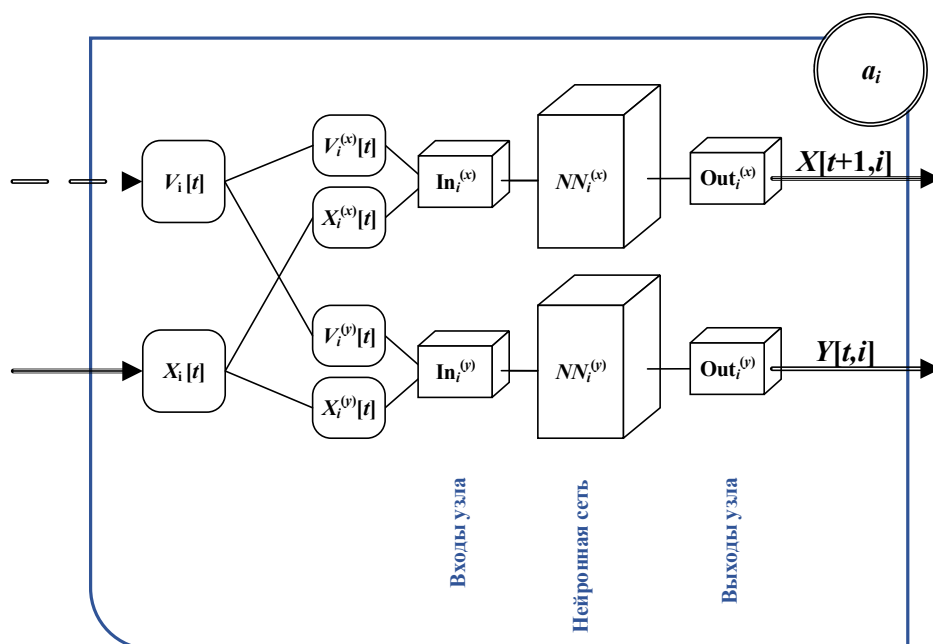


Рис. 4. Общая схема узла ИДНО модели по переходам и выходам

Теоретико-множественный способ

Структура ИДНО модели $N = (A, O)$ является 2-уровневой и содержит множество узлов окрестностной модели $A = \{a_1, a_2, \dots, a_n\}$ и множество окрестностных связей O между ними, где $O = \{O_x, O_v, O_{ier}\}$, $O_x = \bigcup_{i=1}^n O_x[a_i]$ – множество окрестностных связей узлов по состояниям,

$O_x[a_i] = \{O_x^{(x)}[a_i], O_x^{(y)}[a_i]\}$; $O_v = \bigcup_{i=1}^n O_v[a_i]$ – по внешним воздействиям, $O_v[a_i] = \{O_v^{(x)}[a_i], O_v^{(y)}[a_i]\}$;
 $O_{ier} = \{O_{ier}^{(x)}, O_{ier}^{(y)}\}$ – это множество иерархических окрестностных связей между узлами.

Каждый узел модели $a_i = \{NN_i^{(x)}, NN_i^{(y)}\}$ содержит две нейронные сети для переходов X и выходов Y соответственно, где $NN_i^{(x)}$ и $NN_i^{(y)}$ является структурой нейронных сетей для пересчета состояний и выходов модели в узле a_i соответственно. Данные структуры включают в себя:

- 1) количество входных нейронов, которые состоят из количества внешних воздействий и количества состояний;
- 2) количество скрытых слоев, задается в зависимости от задачи;
- 3) количество нейронов на скрытых слоях, задается в зависимости от задачи;
- 4) количество нейронов на выходном слое, которое формируется из количества выходов или состояний;
- 5) тип нейронной сети, выбирается в зависимости от задачи.

Окрестности рассматриваемого примера (на рис. 2 и 3) будут заданы следующим образом:

$$\begin{array}{ll} O_x^{(x)}[a_1] = \{\emptyset\}; & O_v^{(x)}[a_1] = \{a_1\}; \\ O_x^{(x)}[a_2] = \{a_1, a_3, a_4\}; & O_v^{(x)}[a_2] = \{a_2\}; \\ O_x^{(x)}[a_3] = \{\emptyset\}; & O_v^{(x)}[a_3] = \{a_3\}; \\ O_x^{(x)}[a_4] = \{\emptyset\}; & O_v^{(x)}[a_4] = \{\emptyset\}; \\ O_x^{(x)}[a_5] = \{a_4\}; & O_v^{(x)}[a_5] = \{a_5\}; \\ O_x^{(y)}[a_3] = \{a_2\}; & O_v^{(y)}[a_3] = \{a_3\}; \\ O_x^{(y)}[a_4] = \{a_2\}; & O_v^{(y)}[a_4] = \{\emptyset\}; \\ O_x^{(y)}[a_5] = \{a_2, a_4\}; & O_v^{(y)}[a_5] = \{a_5\}. \end{array}$$

Матричный способ

Также ИДНО модели можно представить в виде матриц связей для переходов и выходов:

$$\begin{array}{l} S_x^{(x)} = \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{matrix} \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \dots & s_{n,n} \end{bmatrix}; \end{array} \quad \begin{array}{l} S_v^{(x)} = \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{matrix} \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \dots & s_{n,n} \end{bmatrix}; \end{array}$$

$$\begin{array}{l} S_x^{(y)} = \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{matrix} \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \dots & s_{n,n} \end{bmatrix}; \end{array} \quad \begin{array}{l} S_v^{(y)} = \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{matrix} \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \dots & s_{n,n} \end{bmatrix}, \end{array}$$

где $S_x^{(x)}$ – это матрица смежности для переходов по состояниям, $S_v^{(x)}$ – это матрица смежности для переходов по внешним воздействиям, $S_x^{(y)}$ – это матрица смежности для выходов по состояниям, $S_v^{(y)}$ – это матрица смежности для выходов по внешним воздействиям, $s_{i,j} = 1$, если существует связь между узлом a_i и a_j , $s_{i,j} = 0$, если не существует связь между узлом a_i и a_j .

Для примера, представленного выше (на рис. 2 и 3), матрицы смежности будут задаваться следующим образом:

$$\begin{aligned}
 S_x^{(x)} &= \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}; & S_v^{(x)} &= \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}; \\
 S_x^{(y)} &= \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}; & S_v^{(y)} &= \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}.
 \end{aligned}$$

Описание модели ИДНО

Рассмотрим иерархическую динамическую нейро-окрестностную модель, которая в общем случае задается набором:

$$NS_{IER}^{NN} = (N, X, V, Y, G, F, t_0, X[t_0], t),$$

где:

1) N – структура модели;

2) $X \in R^{\sum_{i=1}^n p_i}$ – блочный вектор состояний окрестностной модели в текущий момент времени:

$$X = (X_1 \quad X_2 \quad \dots \quad X_n)^T,$$

где p_i – количество состояний узла a_i ;

3) $V \in R^{\sum_{i=1}^n m_i}$ – блочный вектор внешних воздействий окрестностной модели в текущий момент времени:

$$V = (V_1 \quad V_2 \quad \dots \quad V_n)^T,$$

где m_i – количество внешних воздействий узла a_i ;

4) $Y \in R^{\sum_{i=1}^n l_i}$ – блочный вектор выходов окрестностной модели в текущий момент времени:

$$Y = (Y_1 \quad Y_2 \quad \dots \quad Y_n)^T,$$

где l_i – количество выходов узла a_i ;

5) G – вектор-функция переходов для пересчета состояний окрестностной модели:

$$G = (G_1 \quad G_2 \quad \dots \quad G_n)^T,$$

$$X[t+1, i] = G_i(X_i^{(x)}[t], V_i^{(x)}[t]),$$

где $t \in T = \{t_0, t_0 + 1, \dots\}$, G_i – это функция нейронной сети пересчета состояний для узла a_i ;

6) F – вектор-функция выходов окрестностной модели:

$$F = (F_1 \quad F_2 \quad \dots \quad F_n)^T,$$

$$Y[t, i] = F_i(X_i^{(y)}[t], V_i^{(y)}[t]),$$

где F_i – это функция нейронной сети выходов для узла a_i ;

7) X^0 – начальное состояние модели в момент времени t_0 :

$$X_i^0 = (X_i^0 \quad X_2^0 \quad \dots \quad X_n^0)^T.$$

Нейронные сети внутри узлов представляют собой следующую модель:

$$\begin{cases} NN_i^{(x)} = (N_i^{(x)}, V_i^{(x)}, Y_i^{(x)}, G_i); \\ NN_i^{(y)} = (N_i^{(y)}, V_i^{(y)}, Y_i^{(y)}, F_i), \end{cases}$$

где $V_i^{(x)} = \{X_i^{(x)}[t], V_i^{(x)}[t]\}$; $V_i^{(y)} = \{X_i^{(y)}[t], V_i^{(y)}[t]\}$ – входы нейронных сетей, которые содержат внешние воздействия V и состояния X соответственно в момент времени t ; $Y_i^{(x)} = X_i[t+1]$, $Y_i^{(y)} = Y_i[t]$ – это выходы нейронных сетей по состояниям и выходам соответственно; G_i – вектор-функция переходов для пересчета состояний окрестностной модели; F_i – вектор-функция для пересчета выходов окрестностной модели.

Идентификация модели

Для выполнения параметрической идентификации модели необходимо подготовить выборку данных. Набор данных содержит следующие факторы:

- 1) входы для нейронной сети $NN_i^{(x)}$: $V_i^{(x)}[t]$ и $X_i^{(x)}[t]$;
- 2) выход для нейронной сети $NN_i^{(x)}$: $X[t+1, i]$;
- 3) входы для нейронной сети $NN_i^{(y)}$: $V_i^{(y)}[t]$ и $X_i^{(y)}[t]$;
- 4) выход для нейронной сети $NN_i^{(y)}$: $Y[t, i]$.

В иерархических динамических нейро-окрестностных моделях расчет состояний и выходов выполняется с учетом связей между окрестностными узлами, в которых помещены нейронные сети.

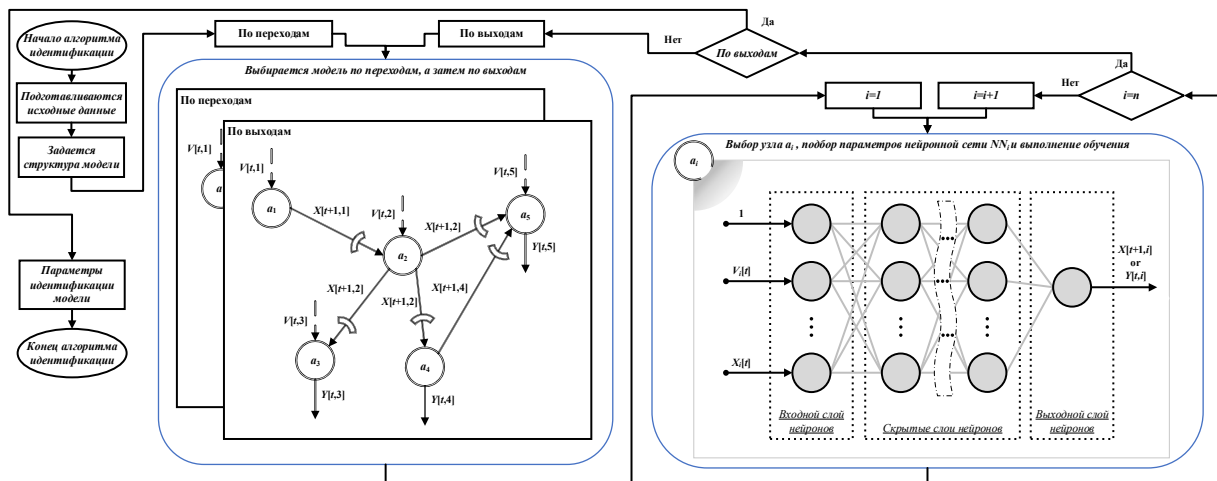


Рис. 5. Схема идентификации ИДНО модели

Опишем алгоритм идентификации ИДНО модели (рис. 5):

1. Начало алгоритма идентификации. Переход к шагу 2.

2. Подготовка исходных данных. На этом этапе происходит анализ и обработка имеющихся данных, относящихся к текущему объекту или процессу. При сборе информации в любой сложной системе так или иначе возникают различные сбои, ошибки ввода и несоответствия. Для повышения достоверности модели необходимо исключить ошибки входных данных, к которым относятся:

- а) противоречивость информации;
- б) наличие пропусков в данных;
- в) аномальные значения;
- г) ошибки, допущенные при вводе данных.

Переход к шагу 3.

3). Задается структура модели, которая определяет:

- а) количество узлов;
- б) связи узлов по состояниям O_x и по внешним воздействиям O_y .

Переход к шагу 4.

4. Выбирается структура модели по переходам.

Переход к шагу 5.

5. Выбирается структура модели: если ранее была по переходам, то сделать по выходам и переход к шагу 6, иначе переход к шагу 7.

6. Выполняется идентификация, заключающаяся в обучении нейронных сетей внутри каждого из узлов окрестностной модели:

- а) выбирается набор связей узлов модели, включающий в себя узлы и заданные связи между ними в соответствии со структурой. Переход к пункту «б» шага 6;
- б) пусть текущий номер узла $i = 1$. Переход к пункту «в» шага 6;
- в) узел a_i для выбранного шаблона. Переход к пункту «г» шага 6;
- г) нейронная сеть обучается в узле a_i для выбранного набора связей узлов. Если $i = n$, то переход к шагу 5, иначе текущий номер узла $i = i + 1$ и перейти к пункту «в» шага 6.

7. Сохраняются параметры идентификации модели для структуры модели по переходам и по выходам. Переход к шагу 8.

8. Конец алгоритма.

Пример построения ИДНО моделей

Для применения модели была разработана программа на языке программирования Python [19–21], которая позволяет обучить нейро-окрестностную модель.

Для того чтобы получить модель объекта или процесса, необходимо подготовить исходные данные в виде Excel-таблицы с достаточным количеством факторов и наблюдений.

Рассмотрим пример ИДНО модели, которая будет обучаться на данных температур и влажности комнат «умного дома», чтобы на их основе прогнозировать общее энергопотребление бытовой техники в доме с учетом отопления и погодных условий.

Набор данных взят с [22] и содержит наблюдения с лагом в 10 мин в течение примерно 4,5 месяца. Условия температуры и влажности в доме отслеживались с помощью соответствующих датчиков в каждой из комнат. Данные об энергопотреблении регистрировались с помощью счетчиков электроэнергии. Данные о погоде взяты с ближайшей метеостанции аэропорта, были загружены из общедоступного набора данных сайта Reliable Prognosis (rp5.ru) и объединены с наборами экспериментальных данных с использованием столбца даты и времени.

Перед обучением модели была проведена подготовка данных, а именно:

1. Проверены данные на корректность.
2. Рассчитана матрица корреляций. На основе данной матрицы были выявлены сильно коррелируемые факторы среди входных воздействий и состояний модели. Конечный вариант используемых факторов и их описание представлен в таблице.

Описание параметров данных

№ п/п	Название фактора	Тип фактора	Описание фактора, ед. измерения
1	date	–	Время замера
2	Appliances	Y2	Приборы потребления, Вт/ч
3	lights	V2_1	Освещение, Вт/ч
4	RH_1	X1_1	Влажность в кухне, %
5	T2	X1_2	Температура в гостиной, °C
6	RH_2	X1_3	Влажность в гостиной, %
7	T3	X1_4	Температура в прачечной, °C
8	RH_5	X1_5	Влажность в ванной комнате, %
9	RH_6	X2_1	Влажность снаружи здания, %
10	RH_9	X1_6	Влажность в спальне родителей, %
11	Press_mm_hg	V2_2	Атмосферное давление, мм рт. ст.
12	RH_out	V2_3	Влажность на улице, %
13	Windspeed	V2_4	Скорость ветра, м/с
14	Visibility	V2_5	Видимость, км
15	Tdewpoint	V2_6	Точка отсчета (от метеостанции), °C

3. Удалены выбросы из данных, применен отбор с помощью межквартильного диапазона.
4. Выполнена кластеризация с последующей категоризацией данных по временному признаку date (по интервалам).
5. Формируется ИДНО модель прогноза энергопотребления, схема представлена на рис. 6.

Количество узлов модели для исследуемого процесса: 2

Количество внешних воздействий: 1

Количество выходов: 1

Количество наблюдений в обучающей выборке: 9904.

Количество наблюдений в валидационной выборке: 4245.

Количество наблюдений в тестовой выборке: 3538.

На рис. 7. показан график изменения валидационной ошибки MAPE на эпохах обучения нейронной сети внутри узла a_2 окрестностной модели по выходам.

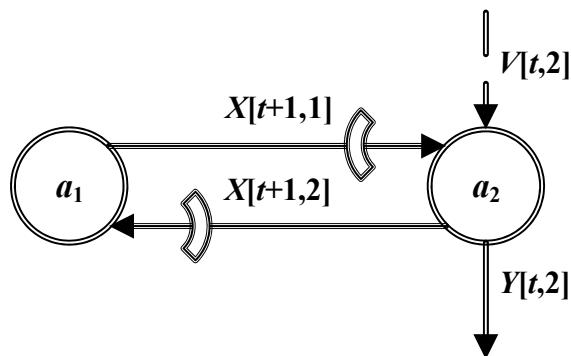


Рис. 6. Схема ИДНО модели прогнозирования энергопотребления

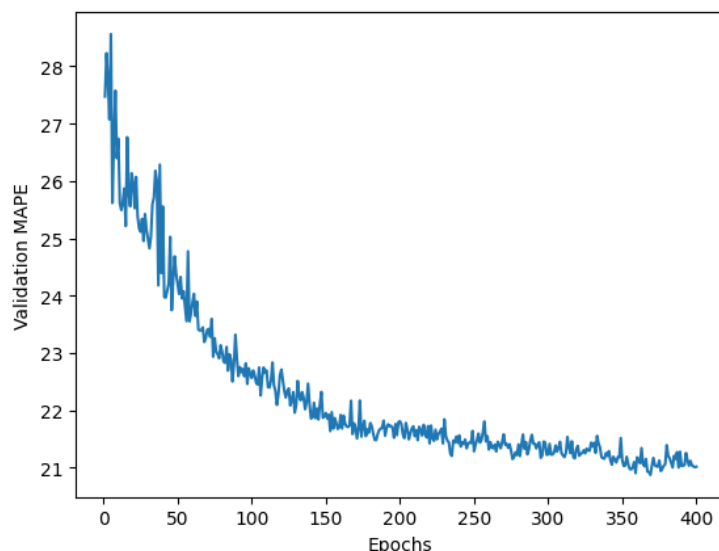


Рис. 7. Валидационная ошибка MAPE на каждой эпохе обучения

Полученная нейронная сеть ИДНО модели обладает параметрами, которые представлены ниже.

Входной слой, количество нейронов соответствует количеству входов 14.

Нейросеть содержит 5 скрытых слоев со следующими характеристиками:

- 1-й скрытый слой: полносвязный, 128 нейронов, функция активации “selu”;
- 2-й скрытый слой: слой прореживания Dropout(0.2);
- 3-й скрытый слой: полносвязный, 64 нейрона, функция активации “relu”;
- 4-й скрытый слой: слой прореживания Dropout(0.2);
- 5-й скрытый слой: полносвязный, 64 нейрона, функция активации “elu”.

Выходной слой состоит из одного нейрона, функция активации “linear”.

Оптимизатор “adam”; функция потерь “huber”; метрика “mare”.

В результате обучения ИДНО модели для каждого из узлов были подобраны оптимальные параметры нейронных сетей. Средняя точность обучения всей модели составляет 79,8 %.

Заключение

Таким образом, в статье выделены основные этапы развития математического моделирования сложных распределенных процессов, в результате которого появился новый класс окрестностных моделей ИДНО модели. Показана обобщенная схема связей классов окрестностных моделей. Отражены способы представления ИДНО моделей: графический, теоретико-множественный, матричный. Описана ИДНО модель и алгоритм ее идентификации. Реализована программа идентификации и автоматического подбора оптимальных параметров структуры нейронных сетей в узлах ИДНО модели на языке программирования Python. В качестве примера обучена ИДНО модель процесса теплообмена в комнатах дома с системой «умный дом». Сделаны выводы по проделанному исследованию.

Список литературы

1. Голубева, Н.В. Математическое моделирование систем и процессов: учеб. пособие / Н.В. Голубева. – СПб.: Лань, 2016. – 192 с.
2. Дворецкий, С.И. Моделирование систем: учебник / С.И. Дворецкий. – М.: Академия, 2019. – 304 с.
3. Яглом, И.М. Математические структуры и математическое моделирование / И.М. Яглом. – М.: Ленанд, 2018. – 144 с.
4. Федоткин, И.М. Математическое моделирование технологических процессов / И.М. Федоткин. – М.: КД Либроком, 2018. – 416 с.
5. Жирков, А.М. Математическое моделирование систем и процессов: учебное пособие / А.М. Жирков, Г.М. Подопригора, М.Р. Цуцунава. – СПб.: Лань КПП, 2016. – 192 с.
6. Юмагулов, М.Г. Введение в теорию динамических систем: учебное пособие / М.Г. Юмагулов. – М.: Лань, 2015. – 272 с.
7. Решмин, Б.И. Имитационное моделирование и системы управления / Б.И. Решмин. – Вологда: Инфра-Инженерия, 2016. – 74 с.
8. Блюмин С.Л. Окрестностные системы / С.Л. Блюмин, А.М. Шмырин // Липец. эколого-гуманитар. ин-т. – Липецк: Липец. эколого-гуманитар. ин-т, 2005. – 131 с.
9. Shang, Y. Multi-agent coordination in directed moving neighborhood random networks / Y. Shang. – Text: electronic // Chinese Physics B. – 2010. – Vol. 19, no. 7. – Article ID 070201.
10. Томилин, А.А. Использование окрестностно-временного моделирования в задачах формирования организационных структур / А.А. Томилин // Управление большими системами. – 2007. – Вып. 18. – С. 91–106.
11. Shmyrin, A. A measure of the non-determinacy of a dynamic neighborhood model / A. Shmyrin, I. Sedykh // Systems. – 2017. – Vol. 5, no. 4. – P. 49. DOI: 10.3390/systems5040049
12. Осовский, С. Нейронные сети для обработки информации: пер. с польского / С. Осовский. – М.: Финансы и статистика, 2002.
13. Редько, В.Г. Эволюция, нейронные сети, интеллект: модели и концепции эволюционной кибернетики / В.Г. Редько. – М.: Ленанд, 2019. – 224 с.
14. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Диалектика, 2019. – 1104 с.
15. Sedykh, I. Introducing dynamic neighborhood models into the educational process / I. Sedykh, V. Istomin // Proceedings – 2021 1st International Conference on Technology Enhanced Learning in Higher Education, TELE 2021 : 1, Lipetsk, 24–25 июня 2021 года. – Lipetsk, 2021. – P. 171–175. DOI: 10.1109/TELE52840.2021.9482496

16. Седых И.А. Реализация программы идентификации иерархических динамических нейро-окрестностных моделей / И.А. Седых, В.А. Истомин // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 12–14 декабря 2022 года / Воронежский государственный университет. – Воронеж: Научно-исследовательские публикации, 2023. – С. 614–620.
17. Седых, И.А. Пример идентификации иерархических динамических нейро-окрестностных моделей с переменными окрестностями / И.А. Седых, В. А. Истомин // Автоматизация процессов управления. – 2023. – № 2(72). – С. 63–70. DOI: 10.35752/1991-2927_2023_2_72_63
18. Sedykh, I. Selection of Optimal Parameters of a Hierarchical Dynamic Neuro-neighborhood Model / I. Sedykh, V. Istomin // 2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – Lipetsk, Russian Federation. – 2023. – P. 255–257. DOI: 10.1109/SUMMA60232.2023.10349470
19. Доусен, М. Программируем на Python (Python Programming for the Absolute Beginner) / М. Доусен // СПб.: Питер, 2016. – 416 с. (Серия «Бестселлеры O'Reilly»).
20. К., Рейтц Автостопом по Python / К. Рейтц, Т. Шлюссер. – СПб.: Питер, 2017. – 336 с. (Серия «Бестселлеры O'Reilly»).
21. Шелудько, В.М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В.М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с.
22. Сервис Kaggle [Электронный ресурс] / Appliances Energy Prediction. – URL: <https://www.kaggle.com/datasets/loveall/appliances-energy-prediction/data> (дата обращения: 04.03.2024).

References

1. Golubeva N.V. Matematicheskoe modelirovanie sistem i processov: ucheb. posobie [Mathematical modeling of systems and processes]. Saint-Petersburg : Lan, 2016, 192 p.
2. Dvoretzky S.I. Modelirovanie sistem [Modeling systems]. Moscow: Academy, 2019. – 304 p.
3. Yaglom I.M. Matematicheskie struktury` i matematicheskoe modelirovanie [Mathematical structures and mathematical modeling]. Moscow: Lenand, 2018, 144 p.
4. Fedotkin I.M. Matematicheskoe modelirovanie texnologicheskix processov [Mathematical modeling of technological processes]. Moscow: KD Librocom, 2018, 416 p.
5. Zhirkov A.M., Podoprighora G.M., Tsutsunawa M.R. Matematicheskoe modelirovanie sistem i processov [Mathematical modeling of systems and processes]. Saint-Petersburg: Lan KPT, 2016, 192 p.
6. Yumagulov M.G. Vvedenie v teoriyu dinamicheskix sistem. Uchebnoe posobie [Introduction to the theory of dynamic systems. Textbook] . Moscow: Lan, 2015, 272 p.
7. Reshmin B.I. Imitacionnoe modelirovanie i sistemy` upravleniya [Simulation modeling and control systems]. Vologda, Infra-Engineering, 2016, 74 p.
8. Blyumin S.L., Shmyrin A.M. Okrestnostny`e sistemy` [Neighborhood systems]. Lipetsk: *Lipetsk environmentalist-humanitarian. Institute*, 2005, 131 p.
9. Shang Y. Multi-agent coordination in directed moving neighborhood random networks. *Chinese Physics B*, 2010, vol. 19, no. 7, article ID 070201.

10. Tomilin A.A. Ispol'zovanie okrestnostno-vremennogo modelirovaniya v zadachax formirovaniya organizacionny'x struktur [Use of neighborhood-time modeling in problems of formation of organizational structures]. *Large Scale Systems Control*, 2007, iss. 18, pp. 91–106.
11. Shmyrin, A., Sedykh I. A measure of the non-determinacy of a dynamic neighborhood model. *Systems*, 2017, vol. 5, no. 4, pp. 49. DOI 10.3390/systems5040049.
12. Osovsky S. Nejronny'e seti dlya obrabotki informacii [Neural networks for information processing]. Moscow: Finance and Statistics, 2002.
13. Redko V.G. E'volyuciya, nejronny'e seti, intellekt: Modeli i koncepcii e'volyucionnoj kibernetiki. [Evolution, neural networks, intelligence: models and concepts of evolutionary cybernetics]. Moscow: Lenand, 2019, 224 p.
14. Khaikin S. Nejronny'e seti: polny'j kurs [Neural networks: a complete course]. Moscow: Dialectics, 2019, 1104 p.
15. Sedykh I., Istomin V. Introducing dynamic neighborhood models into the educational process. *Proceedings of 2021 1st International Conference on Technology Enhanced Learning in Higher Education (TELE)*, 2021, pp. 171-175. DOI: 10.1109/TELE52840.2021.9482496.
16. Sedykh I.A., Istomin V.A. Realizaciya programmy` identifikacii ierarxicheskix dinamicheskix nejro-okrestnostny'x modelej [Implementation of a program for identifying hierarchical dynamic neuro-neighborhood models]. *Current problems of applied mathematics, computer science and mechanics: collection of works of the International scientific conference, Voronezh, December 12–14, 2022 / Voronezh State University*. Voronezh: Scientific research publications, 2023, pp. 614-620.
17. Sedykh I.A., Istomin V.A. Primer identifikacii ierarxicheskix dinamicheskix nejro-okrestnostny'x modelej s peremenny'mi okrestnostyami [An example of identification of hierarchical dynamic neuro-neighborhood models with variable neighborhoods]. *Automation of control processes*, 2023, no. 2(72), pp. 63-70. – DOI 10.35752/1991-2927_2023_2_72_63.
18. Sedykh I., Istomin V. Selection of Optimal Parameters of a Hierarchical Dynamic Neuro-neighborhood Model. *Proceedings of 2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, IEEE, 2023, pp. 255-257. DOI: 10.35752/1991-2927_2023_2_72_63.
19. Dawsen M. Programmiruem na Python [Python Programming for the Absolute Beginner]. Ed. "Peter", O'Reilly Bestsellers series, 2016, 416 p.
20. Reitz K., Schlusser T. Avtostopom po Python [The Hitchhiker's Guide to Python]. Saint-Petersburg: Peter, 2017, 336 p.
21. Sheludko, V.M. Yazyk programirovaniya vy`sokogo urovnya Python. Funkcii, struktury` danny'x, dopolnitel'ny'e moduli [High-level programming language Python. Functions, data structures, additional modules]. Rostov-on-Don, Taganrog, Southern Federal University Publishing House, 2017, 107 p.
22. Kaggle, available at: <https://www.kaggle.com/datasets/loveall/appliances-energy-prediction/data> (Accessed 04 March 2024).