DOI: 10.15514/ISPRAS-2023-35(5)-6

The Open System for Storing and Processing of a Dataset of Combinational Circuits

D.A. Miachin, ORCID: 0009-0009-4538-5286 <*danchikmiachin@gmail.com*>
V.P. Pugach, ORCID: 0009-0003-3260-1011 <*vppugach@edu.hse.ru*>
S.S. Avdeiuk, ORCID: 0009-0000-9183-4497 <*ssavdeyuk@edu.hse.ru*>
V.V. Zunin, ORCID: 0000-0002-9117-4879 <*vzunin@hse.ru*>
A.Y. Romanov, ORCID: 0000-0002-9410-9431 <*a.romanov@hse.ru*>

HSE University,
20, Myasnitskaya st., Moscow, 101000, Russia.

Abstract. This paper presents an open-source software for generation, storage, and analysis of combinational circuits. The previously created methods for generating combinational circuits have been optimized, and a dataset has been formed. The generation of combinational circuits is carried out on various devices. The application implements the possibility to combine the generated datasets into a single storage (Synology Drive), as well as analyze the fault tolerance of combinational circuits using various methods for their evaluation. New possible methods for assessing combinational circuits' reliability using machine learning are proposed.

Keywords: combinational circuits; dataset; reliability; machine learning; client-server application.

For citation: Miachin D.A., Pugach V.P., Avdeiuk S.S., Zunin V.V., Romanov A.Y. The open system for storing and processing of a dataset of combinational circuits. *Trudy ISP RAN/Proc. ISP RAS,* vol. 35, issue 5, 2023. pp. 81-90. DOI: 10.15514/ISPRAS-2023-35(5)-6.

Открытая система хранения и обработки набора данных комбинационных схем

Д.А. Мячин, ORCID: 0009-0009-4538-5286 <*danchikmiachin@gmail.com*>
В.П. Пугач, ORCID: 0009-0003-3260-1011 <*vppugach@edu.hse.ru*>
С.С. Авдеюк, ORCID: 0009-0000-9183-4497 <*ssavdeyuk@edu.hse.ru*>
В.В. Зунин, ORCID: 0000-0002-9117-4879 <*vzunin@hse.ru*>
А.Ю. Романов, ORCID: 0000-0002-9410-9431 <*a.romanov@hse.ru*>

Национальный исследовательский университет «Высшая школа экономики»,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. В этой статье представлено программное обеспечение с открытым исходным кодом для генерации, хранения и анализа комбинационных схем. Оптимизированы созданные ранее методы генерации комбинационных схем и сформирован датасет. Генерация комбинационных схем может осуществляться на различных устройствах. В приложении реализована возможность объединения сгенерированных наборов данных в единое хранилище (Synology Drive), а также анализа отказоустойчивости комбинационных схем с использованием различных методов их оценки. Предложены новые возможные методы оценки надежности комбинационных схем с использованием машинного обучения.

Ключевые слова: комбинационные схемы; датасет; отказоустойчивость; машинное обучение; клиент-серверное приложение.

Для цитирования: Мячин Д.А., Пугач В.П., Авдеюк С.С., Зунин В.В., Романов А.Ю. Открытая система хранения и обработки набора данных комбинационных схем. Труды ИСП РАН, том 35, вып. 5, 2023 г., стр. 81–90 (на английском языке). DOI: 10.15514/ISPRAS–2023–35(5)–6.

1 Introduction

A combinational circuit [1] is a digital circuit that has no memory and (depending on the discrete signals received at the inputs) produces unambiguously defined logic signals at the outputs [2].

In any modern electronic devices, both memory circuits and combinational circuits are used. And, like any other electronic devices, combinational circuits are subject to possible failures and malfunctions. Failure of the circuit can lead to malfunction of the whole device, which (in turn) can lead to serious consequences. That is why fault tolerance of the combinational circuit is one of its key characteristics and indicators of its quality affecting the operation of various electronic devices.

Due to the ubiquitous use of digital circuits, the study and creation of methods for their generation, as well as the estimation of the parameters of such circuits (including fault tolerance [3]), is a relevant topic for scientific research and is of great importance for the development of modern technologies. Thus, there is a need to create open-source software that will allow generating large amounts of data for conducting research on predicting the parameters of combinational circuits using machine learning methods.

Existing fault tolerance evaluation methods (e.g., [3–7]) have a number of drawbacks and problems, which will be described in detail in this paper; optimization of these methods and creation of new ones are relevant at present.

2 Problem statement

Currently, there are no client-server applications for users to conveniently generate large datasets of combinational circuits on various devices with calculation of their basic parameters, while having only an Internet access or connecting computing and storage devices locally. Often, this task requires the installation of specialized software. Also, currently developed programs for generating datasets of combinational circuits [8–10] do not have functionality for visualizing the generated circuits and assessing their fault tolerance, which is the key in analyzing the quality of the circuit.

Thus, the purpose of this study is to develop a unified system for storing and processing a dataset of digital circuits to provide a convenient user interaction with a program for generating circuits using a web interface. The created software provides a user with the ability to select an algorithm for generating combinational circuits, configure the necessary parameters for generation, and also reduce the time for analyzing the fault tolerance of combinational circuits and their other parameters in the future. The developed client-server application allows evaluating the fault tolerance of the generated combinational circuits directly from the application without requiring any installation of additional software, as well as visualize the generated combinational circuits.

One of the additional problems being solved in the course of the study was the problem of optimizing the previously developed software [11] used as the basis for the study – to speed up its work and improve its convenience. The software was also translated (by manual transpilation into C++) from the C# language.

3 Analog analysis

The search and analysis of existing applications and programs for generating combinational circuits showed that at present there are no full-fledged analogues that allow generating combinational circuits using various generation algorithms and estimating their parameters; algorithms for constructing combinational circuits based on truth tables and trees are mainly used.

Consider the programs capable of working with virtual (emulated) combinational circuits: Multisim Workbench [9] and Logicly [10].

To use Multisim Workbench [9], one must manually (graphically) enter the logic elements of each circuit and manually create the circuits themselves. Due to this implementation, the process of placing circuit components becomes more complicated than using code or automated creation. Transferring data in text form to a graphics window complicates program development. Thus, their advantages, the ability to visualize circuits becomes a disadvantage, if automated generation of combinational circuits is required. Also, a big disadvantage of Multisim Workbench is that it is a closed commercial program with a rather high license price.

The main advantage of Logicly [10] is a fairly low entry threshold and a convenient graphic design, but it is limited in that it allows modeling a circuit with only a small number of elements. The program cannot generate combinational circuits based on the given generation parameters, but only allows manual creation of circuits.

Thus, the lack of automation and any integrated means of storing and processing the results of work do not allow the full use of existing solutions for working with combinational circuits; so, the development of new ones is needed.

4 Software architecture

In the course of this study, a client-server application for generating combinational circuits and their aggregation in the selected storage was developed. The client-server architecture was chosen, since this is what will allow setting up several server parts for simultaneous generation of circuits on several devices, which will significantly speed up the circuit generation. Also, if necessary, the server parts can be accessed remotely for ease of interaction.

To provide a simplified procedure for adding new circuit generation methods, a special structure for the interaction of generation methods with the dataset generator was developed. The architecture of the developed client-server application consists of two main components (Fig. 1), the client and the server, interacting with each other using the API.

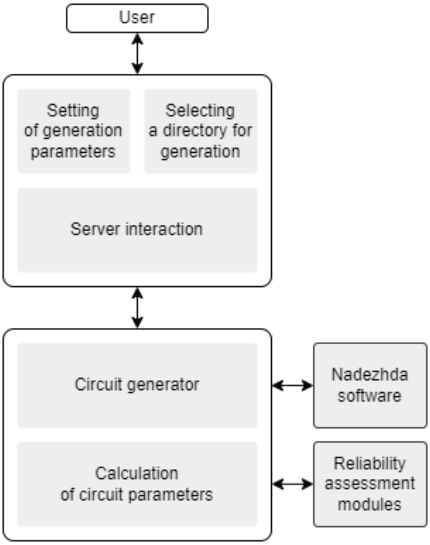
User
Setting of generation parameters
Selecting a directory for generation
Server interaction
Circuit generator
Nadezhda software
Calculation of circuit parameters
Reliability assessment modules

Fig. 1. Architecture of the developed application.

The server can run a generator module associated with various generation methods. Thus, when adding new methods, the principle of the program does not change, and the user and developers have the opportunity to asynchronously modify the software to suit their needs without making large-scale changes to the program. The main advantage of this architecture is the flexibility for future improvements.

The frontend part allows the user to configure parameters for generating combinational circuits and select a directory for their aggregation, while the backend part calls a function to generate combinational circuit datasets, performs visualizations of the generated circuits, and also calculates various circuit parameters, including their fault tolerance using various reliability assessment methods.

The first and main advantage of the developed software is the ability not only to generate combinational circuits, but also to evaluate their fault tolerance within one software, as well as modify it for other tasks.

The second advantage of the developed software is that the client-server application allows using it without in-depth knowledge in the field of programming and circuitry.

The third advantage is that the developed software allows automating the process of filling in the parameters for generation; at the same time, the functionality of automatic visualization of the generated circuits is implemented.

The fourth advantage is the availability of aggregation of generated circuits in one place. The developed generation program allows saving the generated datasets of combinational circuits in the Verilog format, as well as saving the information about each circuit in the JSON format [12]. Also, in the folder with each circuit, there is a PNG visualization generated using the Yosys software [13]. All generated circuits are aggregated in the given storage, for which Synology Drive [14] was selected. It was chosen due to its simple library for interacting with the storage, as well as ease of use.

4.1 Web-component

A web application which interacts with the program for generating combinational circuit datasets and displays the generation results to the user on the site using the server is implemented. The user has the ability to set generation parameters that are stored in the database on the server. The web interface also prevents the user from entering invalid values and controls limits on the generation parameters, depending on the chosen generation method.

After the user sets all the parameters necessary for generating, a JSON request with these parameters is generated; it is sent to the server, which (upon receiving requests from the client) launches the combinational circuit dataset generator for these parameters and returns a link to the location of the generated datasets.

During the generation process, the user can track the progress of the generator program using progress bars. At the end of the program, the web interface receives a request from the server containing a link to the generated datasets divided into different folders (depending on the generation method) and also receives the parameters of the generated datasets, such as the number of circuit inputs and outputs.

Combination circuits are generated asynchronously on the server, which eliminates the need for the user to keep any program or site running. Thus, the user does not have to worry about their safety, since the service is available wherever there is an Internet connection. At the same time, installation of additional software is also not required, as all the interaction takes place in a web browser.

Using a server with generation, there is a potential opportunity to use the developed service from various devices on which each user has a unique account with unique generation data. As a result, all generated circuits will be aggregated in a single remote folder. That is, as a result of creating a web application, the basis for the future project scaling or further integration into other applications is formed.

To create the service, all modern methods applied in web development were used:

- the development was done using the React JS framework;
- the possibility of obtaining up-to-date data and data updating (during the operation of the site using asynchronous requests) was implemented.

4.2 Server component

The server component of the application receives and transmits data from both the web application and the generation program. Interaction with the web application takes place using the API. This is a modern interface for communication between backend and frontend components.

The interaction of the server component with the generation program is carried out using sockets. This interface was chosen because it allows programs developed in different programming languages to exchange data, which increases the convenience of further development. The server is implemented in Python3 using the Django REST framework, while the circuits generator is implemented in C++. This is done in order to be able to use for generation the programs developed in other programming languages, not changing the server part.

For a more convenient demonstration of the generation result to the user, the capability to visualize the generated combinational circuits using the Yosys software [13] was added to the server.

Yosys software creates DOT files from Verilog files generated by the program. Next, using the Graphviz library [15] a schema file in png format is generated from a DOT format file, which has a human-readable form (Fig. 2).

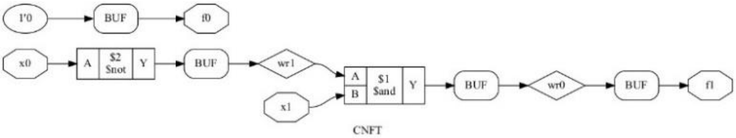

Fig. 2. Example of a generated combinational circuit.

Thus, the user can analyze the graphical interpretation of the circuit, which is located in the directory selected for generation next to the Verilog format file.

The possibility to run the generation program and the server on the same computer is Implemented. Due to the adopted engineering solutions, this does not cause additional problems, unlike the implementation method using system interrupts. There are enough ports on the computer to transfer the data via sockets; so, running programs on one device is not difficult. It also allows scaling the program by running many back-ends for parallel generation of combinational circuits.

4.2.1 Optimization of the generation algorithms

The program for generating circuits allows generating a dataset to study their fault tolerance. In the case of conventional prediction methods, the fault tolerance of the circuits generated is compared with an ideal value, and the predictive power of the new methods is evaluated. It is also possible to use the datasets generated to train standard Machine Learning models.

When training neural networks, a large data set allows (in addition to predicting fault tolerance) generating new circuits corresponding to the same logical function but with greater fault tolerance.

Different methods of generating combinational circuits provide for the presence of different necessary parameters. From the point of view of machine learning, this is useful, as it avoids the homogeneity of circuits and prevents the occurrence of the retraining factor for some specific type of generation.

Four types of generation are currently implemented [16]:

1) In truth tables (Combinational Circuit Generation using Random Truth Tables; CCGRTT).
2) By specifying the number of circuit levels (Combinational Circuits Generation by the Random Connection of Gates; CCGRCG);
3) By setting the number of logical operators in the circuit (Combinational Circuits Generation by Random Vertex Connection; CCGRVC);
4) By using a genetic algorithm (Combinational Circuits Generation based on Genetic Algorithm; CCGGA);

5) By using custom circuit and their post-processing (Combinational Circuits Generation based on the User-Defined Schemes, CCGUDS).

All these generation methods do not correlate with each other and allow obtaining a variety of data. The circuits generated by these generation methods are built on the basis of the implemented directed graph class, which makes it possible to conveniently interact with them inside the program.

CCGRTT method is one of the simplest. For this generation method, the user must specify the desired number of inputs and outputs for the circuit, as well as the method for minimizing logical functions for logical expressions (PDNF and/or PCNF).

CCGRCG generates a combinational circuit depending on the limit on the number of levels (the tree depth) specified by the user and on the number of elements at each level.

CCGRVC generates combinational circuits depending on the limit on the number of logical blocks, which is set by the user. The user specifies the number of different logic elements to be present in the circuit. Circuits are generated based on these parameters.

The genetic algorithm [17] is one of the promising methods. It lies in the fact that the program generates a starting population, depending on the type of chromosomes specified by the user. The initial population is then checked against the stopping criterion. If the requirements are not met, in the resulting population, parents are selected, which are then crossed with each other as a result of which mutations occur. Then there is a new population selection among the descendants. The cycle continues until the stop criterion is reached.

CCGUDS gives the possibility to manually add combinational circuits in Verilog format for their subsequent processing and adding to the final dataset. This method makes it possible to manually supplement the dataset with the necessary circuits, as well as check the operability of the software as it is being finalized.

The generation algorithms are described in more detail in [16]. In this study, they were finalized and optimized to speed up their work using the C ++ language. Using C++ allows making the program cross-platform since C# is focused on the Windows operating system. Also, the advantage of C++ is that in tasks that require processing a large amount of data, such as generating large datasets, it can be solved with fewer resources unlike C#. In addition, there are more libraries in C++ to simplify the development.

The algorithms for generating combinational circuits were also improved by adding validation of input data in each algorithm, not just in the user interface. This prevents any errors that might occur when using generation methods separate from the server side. Thus, all necessary restrictions on the user-entered parameters are displayed directly in the web interface. Thus, the user immediately receives information about the correctness of the data and has the opportunity to correct errors before sending a request for the generation.

A support for launching the software from the command line with specified parameters, which allows running it on the server is implemented.

4.2.2 Reliability calculation

The calculation of the fault tolerance of combinational circuits is carried out using the Nadezhda software [18], as well as using the software implementation of the algorithms described in various works [3–7]. They use various methods, ranging from probabilistic ones [3], [6] to methods using machine learning. This makes it possible to compare the presented algorithms and obtain more complete data for the further research.

4.2.3 Unit tests

To simplify the development process, a CMake [19] file was developed the presence of which is a necessary requirement for using the program. The project uses exactly this build system since it is quite simple for the user and allows not thinking about dependencies because they are described in the build files and are installed automatically. This was an additional advantage over the previous version of the C# program.

Also, in the same build system, a separate build of regression tests is implemented, which greatly simplifies the work of developers.

CI/CD is configured in the project repository, which, each time a new version of the project is loaded, builds an application for generation, and also launches all the tests. This way the developers of the project will be notified if incorrect changes are accidentally sent to the server (which sometimes happens due to human error).

5 Approbation

Using the improved software, a dataset (about 30 thousand circuits) needed for further research in the development of methods for assessing the fault tolerance of microelectronic circuits was generated. The dataset contains a variety of combinational circuits due to the non-correlation of the algorithms used for generation and the diversification of parameters for generation. The non-correlation of generation methods makes it possible to avoid retraining a neural network for a specific type of generation, which can adversely affect the universality of the fault tolerance methods developed.

As is shown in work [16], the use of machine learning methods, neural networks or XGBoost [20] makes it possible to make fault tolerance predictions with high accuracy. But in the work, only the parameters of the combinational circuits were used for training and not the circuits themselves, which leads to the fact that several circuits can be mapped to the same parameters with different fault tolerance values. To improve the quality of fault tolerance predictions or other parameters, consider other approaches that can be involved to use the generated dataset for application with machine learning methods.

The first possible use of a dataset is based on the fact that a combinational circuit is basically a graph. Thus, a graph neural network (GNN) can be used, which is a specialized neural network that takes a graph as input. In [21], modern GNNs are divided into four categories: recurrent GNNs, convolutional GNNs, graph autoencoders, and spatial-temporal GNNs. Using convolutional GNNs, one can, for example, make combinational circuit fault tolerance predictions based on a generated dataset or other known parameters. Another example is the use of graph autoencoders to improve the characteristics of circuits (fault tolerance, delays, etc.), which is also possible using the generated dataset with its additional processing and grouping based on the required parameters. GNNs can also be used for the tasks of generating combinational circuits (GAN [22]).

The second application of the dataset created is the use of natural language processing (NLP) methods [23]. Currently, NLP is used to perform various tasks, such as speech recognition [24], text classification [25], named entity extraction, etc. Since the combinational circuit in the dataset is presented as source code in the Verilog language, such text can be processed by NLP methods to classify the combinational circuits into separate categories, for example, by fault tolerance or size, as well as to generate combinational circuits through their Verilog descriptions.

The third possible way to use the generated dataset is to represent the combinational circuit in the form of graph. In [26], a similar solution showed good results for graph classification problems is described. Based on this, we can conclude that this approach can be extended to the problem of classifying combinational circuits and assessing their fault tolerance.

6 Conclusion

Thus, the result of this study is a new open client-server application that allows generating combinational circuits using various generation methods and aggregating them in a selected location. Based on it, a system that allows generating, storing, and processing datasets of microelectronic circuits on various devices was implemented. The program also provides functionality for assessing the fault tolerance of microelectronic circuits and for visualizing the generated circuits using other open Yosys software.

Previously developed methods for assessing fault tolerance were analyzed, and some of them were also implemented. A CMake file was developed, and a CI/CD development method was organized to optimize the software. This increased the reliability of making changes to the program. The software is now organized on the principles of client-server systems. A web interface was developed to simplify user interaction with the program. Restriction checks applied to generation parameters and validation of parameters entered by the user were realized.

During the approbation of software, an extensive dataset of combinational circuits was generated, and an analysis of various machine learning methods that can be used was made.

References

[1]. Harris S. L., Harris D. M. Digital Design and Computer Architecture: RISC-V Edition. Morgan Kaufmann, 2021. 592 p.

[2]. Roy S., Tilak C. T. On Synthesis of Combinational Logic Circuits. International Journal of Computer Applications, vol. 127, no. 1, 2015, pp. 21–26. DOI: 10.5120/IJCA2015906311.

[3]. Han J., Chen H., Boykin E., Fortes J. Reliability evaluation of logic circuits using probabilistic gate models. Microelectronics Reliability, vol. 51, no. 2, 2011, pp. 468–476. DOI: 10.1016/j.microrel.2010.07.154.

[4]. Choudhury M. R., Mohanram K. Reliability Analysis of Logic Circuits. IIEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 3, 2009, pp. 392–405. DOI: 10.1109/TCAD.2009.2012530.

[5]. Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Тельпухова Н. В. Исследование вероятностных методов оценки логической уязвимости комбинационных схем. Проблемы разработки перспективных микро- и наноэлектронных схем, №4, 2016, сс. 121-126. / Stempkovskiy A.L., Telpukhov D.V., Soloviev R.A., Telpukhova N.V. Probabilistic methods for reliability evaluation of combinational Circuits. Problems of Perspective Micro- and Nanoelectronic Systems Development, no. 4, 2016, pp. 121-126 (in Russian).

[6]. Тельпухов Д. В., Соловьев Р. А., Тельпухова Н. В., Щелоков А. Н. Оценка параметра логической чувствительности комбинационной схемы к однократным ошибкам с помощью вероятностных методов. Известия ЮФУ. Технические науки, №7, 2016, сс. 149–158. DOI: 10.18522/2311-3103-2016-7-149158. / Telpukhov D.V., Soloviev R. A., Telpukhova N. V., Schelokov A.N. Ocenka parametra logicheskoj chuvstvitel'nosti kombinacionnoj shemy k odnokratnym oshibkam s pomoshh'ju verojatnostnyh metodov. Izvestija JuFU. Tehnicheskie nauki, no. 7, 2016, pp. 149-158. DOI: 10.18522/2311-3103-2016-7-149158 (in Russian).

[7]. Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В., and Тельпухова Н. В., “Разработка технологически независимых метрик для оценки маскирующих свойств логических схем,” Вычислительные технологии, том 21, №2, 2016, сс. 53–62. / Stempkovskiy A.L., Telpukhov D.V., Soloviev R.A., Myachikov M. V., Telpukhova N. V. The development of technology-independent metrics for evaluation of the masking properties of logic, vol. 21, no. 2, 2016, pp. 53-62.

[8]. Icarus Verilog, Available at: https://github.com/steveicarus/iverilog, accessed 27.07.2023.

[9]. NI Multisim Circuit Design Suite, Available at: https://www.studica.com/NI-Circuit-Design-Suite-Student-Edition-Download, accessed 27.07.2023.

[10]. Logicly, Available at: https://logic.ly/, accessed 27.07.2023.

[11]. CAD_Combinational_Circuits, Available at: https://github.com/RomeoMe5/CAD_Combinational_Circuits, accessed 27.07.2023.

[12]. Json.NET – Newtonsoft, Available at: https://www.newtonsoft.com/json, accessed 27.07.2023.

[13]. Yosys Open Synthesis Suite, Available at: https://yosyshq.net/yosys/, accessed 27.07.2023.

[14]. Synology Drive API, Available at: https://github.com/zbjdonald/synology-drive-api, accessed 27.07.2023.

[15]. Graphviz, Available at: https://graphviz.org/, accessed 28.07.2023.

[16]. Zunin V. V., Romanov A. Y., Solovyev R. A. Developing Methods for Combinational Circuit Generation. In Proceedings - 2022 International Russian Automation Conference (RusAutoCon), 2022, pp. 842–846. DOI: 10.1109/RUSAUTOCON54946.2022.9896390.

[17]. Banzhaf W., Nordin P., Keller R., Francone F. Genetic programming: an introduction on the automatic evolution of computer programs and its applications. Morgan Kaufmann Publishers, 1998.

[18]. Nadezhda - Reliability Enhancement Logic tool for Integrated Circuits design, Available at: https://alphachip-tools.ru/nadezhda.php, accessed 27.07.2023.

[19]. CMake, Available at: https://cmake.org/, accessed 28.06.2023.

[20]. Chen T., Guestrin C. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[21]. Wu Z., Pan S., Chen F., Long G., Zhang C., Yu P. S. A Comprehensive Survey on Graph Neural Networks. IEEE Trans. Neural Networks Learn. Syst., vol. 32, no. 1, 2021, pp. 4–24, DOI: 10.1109/TNNLS.2020.2978386.

[22]. Зунин В.В., Романов А.Ю. Intel OpenVINO™ Toolkit: анализ производительности выполнения генеративно-состязательных нейронных сетей. Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС), выпуск 2, 2021, сс. 83–90. DOI: 10.31114/2078-7707-2021-2-83-90. / Zunin V.V., Romanov A. Y. Intel OpenVINO™ Toolkit: Performance Analysis of Generative Adversarial Neural Networks. Problems of Perspective Micro- and Nanoelectronic Systems Development, issue 2, 2021, pp. 83-90. DOI: 10.31114/2078-7707-2021-2-83-90.

[23]. Otter D. W., Medina J. R., Kalita J. K. A Survey of the Usages of Deep Learning for Natural Language Processing. IEEE Trans. NEURAL NETWORKS Learn. Syst., vol. 32, no. 2, 2021, DOI: 10.1109/TNNLS.2020.2979670.

[24]. Kamath U., Liu J., Whitaker J. Deep Learning for NLP and Speech Recognition. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-14596-5.

[25]. Romanov A., Kozlova E., Lomotin K. Application of NLP Algorithms: Automatic Text Classifier Tool. In Communications in Computer and Information Science, vol. 859, 2018, pp. 310–323. DOI: 10.1007/978-3-030-02846-6_25.

[26]. Tixier A. J. P., Nikolentzos G., Meladianos P., Vazirgiannis M. Graph Classification with 2D Convolutional Neural Networks. Lecture Notes in Computer Science, vol. 11731, 2017, pp. 578–593, DOI: 10.1007/978-3-030-30493-5_54.

Information about authors / Информация об авторах

Данил Александрович МЯЧИН – студент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: машинное обучение, разработка САПР.

Danil Aleksandrovich MIACHIN – student from the HSE University. Research interests: machine learning, CAD development.

Виктория Павловна ПУГАЧ – студент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: машинное обучение, разработка САПР.

Viktoriia Pavlovna PUGACH – student from the HSE University. Research interests: machine learning, CAD development.

Степан Сергеевич АВДЕЮК– студент национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: машинное обучение, разработка САПР.

Stepan Sergeevich AVDEIUK– student from the HSE University. Research interests: machine learning, CAD development.

Владимир Викторович ЗУНИН – аспирант департамента компьютерной инженерии национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: машинное обучение, системы автоматизированного проектирования.

Vladimir Viktorovich ZUNIN – postgraduate student from the HSE University. Research interests: machine learning, CAD systems.

Aleksandr Yur’yevich ROMANOV – PhD in Technical Sciences, Associate Professor, Head of the Laboratory of Computer-Aided Design Systems at the HSE University. Research interests: networks-on-chip, systems-on-chip, machine learning, CAD systems.

Александр Юрьевич РОМАНОВ – кандидат технических наук, доцент, заведующий лабораторией систем автоматизированного проектирования национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: сети на кристалле, системы на кристалле, машинное обучение, системы автоматизированного проектирования.