

ABSTRACT AND REFERENCES

INFORMATION TECHNOLOGY. INDUSTRY CONTROL SYSTEMS

DOI: 10.15587/1729-4061.2026.357361

IMPROVEMENT OF INFORMATION TECHNOLOGY FOR SYNTHESIZING PARALLEL-STREAM STRUCTURES FOR VERTICAL-GROUP COMPUTING OF MULTI-OPERAND NEURAL OPERATIONS IN REAL TIME (p. 6–23)**Ivan Tsmots**

Lviv Polytechnic National University, Lviv, Ukraine

ORCID: <https://orcid.org/0000-0002-4033-8618>**Vasyl Teslyuk**

Lviv Polytechnic National University, Lviv, Ukraine

ORCID: <https://orcid.org/0000-0002-5974-9310>**Yurii Opotyak**

Lviv Polytechnic National University, Lviv, Ukraine

ORCID: <https://orcid.org/0000-0001-9889-4177>**Bohdan Shtohrinets**

Lviv Polytechnic National University, Lviv, Ukraine

ORCID: <https://orcid.org/0009-0001-4956-3862>

This study considers an approach to improving the information technology of structure synthesis for real-time multi-operand neural operation data processing. The task addressed relates to the lack of a formalized approach to the synthesis of such structures that would simultaneously take into account the parameters of data flows, the depth of the pipeline, the degree of parallelism, and hardware limitations while ensuring the specified time characteristics.

Methods for parallel vertical-group calculation of the scalar product, sum of squared differences, and search for maximum and minimum values have been devised. A method for concretizing flow graphs has been improved. Basic parallel-stream computing structures and analytical expressions for estimating hardware costs, pipeline cycle time, and efficiency of hardware resource use have been developed. Based on the above, the information technology for synthesizing parallel-stream structures for vertical-group calculation of real-time multi-operand neural operations has been improved. The task set was solved by combining vertical and group parallelism, conveyerization, modular organization, matching the intensity of data input with the intensity of their processing and a gradual transition from algorithmic description to hardware implementation.

The improved information technology provides a reduction in hardware costs, increased throughput, reduced latency, and the selection of optimal parameters of structures. Simultaneous processing of groups of bit slices reduces the number of pipeline steps, and the specification of flow graphs makes it possible to adapt the structure of calculations to real-time requirements.

In practice, the results could be used for synthesizing specialized FPGA-, ASIC-, SoC-tools for neural-oriented real-time systems with specified characteristics.

Keywords: flow graphs, real-time systems, hardware costs, resource efficiency.

References

- Izonin, I., Tkachenko, R., Hovdysh, N., Berezsky, O., Yemets, K., Tsmots, I. (2025). Cascade-Based Input-Doubling Classifier for Predicting Survival in Allogeneic Bone Marrow Transplants: Small Data Case. *Computation*, 13 (4), 80. <https://doi.org/10.3390/computation13040080>
- Tsmots, I., Teslyuk, V., Łukaszewicz, A., Lukashchuk, Y., Kazymyra, I., Holovatyy, A., Opotyak, Y. (2023). An Approach to the Implementation of a Neural Network for Cryptographic Protection of Data Transmission at UAV. *Drones*, 7 (8), 507. <https://doi.org/10.3390/drones7080507>
- Juracy, L. R., Garibotti, R., Moraes, F. G. (2023). From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks. *Foundations and Trends® in Electronic Design Automation*, 13 (4), 270–344. <https://doi.org/10.1561/10000000060>
- Deng, B. L., Li, G., Han, S., Shi, L., Xie, Y. (2020). Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proceedings of the IEEE*, 108 (4), 485–532. <https://doi.org/10.1109/jproc.2020.2976475>
- Mohaidat, T., Khalil, K. (2024). A Survey on Neural Network Hardware Accelerators. *IEEE Transactions on Artificial Intelligence*, 5 (8), 3801–3822. <https://doi.org/10.1109/tai.2024.3377147>
- Tsmots, I., Teslyuk, V., Kryvinska, N., Skorokhoda, O., Kazymyra, I. (2022). Development of a generalized model for parallel-streaming neural element and structures for scalar product calculation devices. *The Journal of Supercomputing*, 79 (5), 4820–4846. <https://doi.org/10.1007/s11227-022-04838-0>
- Xie, Y., Oniga, S. (2024). A Comprehensive Review of Hardware Acceleration Techniques and Convolutional Neural Networks for EEG Signals. *Sensors*, 24 (17), 5813. <https://doi.org/10.3390/s24175813>
- Wang, Z. (2025). Accelerating Transformer Models: FPGA-Based Hardware Optimization and Heterogeneous Computing Strategies. *Applied and Computational Engineering*, 138 (1), 86–92. <https://doi.org/10.54254/2755-2721/2025.21360>
- Zeng, K., Ma, Q., Wu, J. W., Chen, Z., Shen, T., Yan, C. (2022). FPGA-based accelerator for object detection: a comprehensive survey. *The Journal of Supercomputing*, 78 (12), 14096–14136. <https://doi.org/10.1007/s11227-022-04415-5>
- Tsmots, I., Teslyuk, V., Opotyak, Y., Mamchur, T., Oliinyk, O. (2025). Synthesis of recursive-type neural elements with parallel vertical-group data processing. *Eastern-European Journal of Enterprise Technologies*, 3 (2 (135)), 6–16. <https://doi.org/10.15587/1729-4061.2025.329139>
- Geng, S., Wang, Z., Liu, Z., Zhang, M., Zhu, X., Dan, Y. (2025). Hardware implementation of FPGA-based spiking attention neural network accelerator. *PeerJ Computer Science*, 11, e3077. <https://doi.org/10.7717/peerj-cs.3077>
- Carpegna, A., Savino, A., Carlo, S. D. (2025). Spiker+: A Framework for the Generation of Efficient Spiking Neural Networks FPGA Accelerators for Inference at the Edge. *IEEE Transactions on Emerging Topics in Computing*, 13 (3), 784–798. <https://doi.org/10.1109/tetc.2024.3511676>
- Gong, Y., Xu, Z., He, Z., Zhang, W., Tu, X., Liang, X., Jiang, L. (2022). N3H-Core. *Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 112–122. <https://doi.org/10.1145/3490422.3502367>
- Tsai, T.-H., Ho, Y.-C., Sheu, M.-H. (2019). Implementation of FPGA-based Accelerator for Deep Neural Networks. *2019 IEEE 22nd Interna-*

- tional Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), 1–4. <https://doi.org/10.1109/ddecs.2019.8724665>
15. Kachris, C. (2025). A Survey on Hardware Accelerators for Large Language Models. *Applied Sciences*, 15 (2), 586. <https://doi.org/10.3390/app15020586>
 16. Silvano, C., Ielmini, D., Ferrandi, F., Fiorin, L., Curzel, S., Benini, L. et al. (2025). A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms. *ACM Computing Surveys*, 57 (11), 1–39. <https://doi.org/10.1145/3729215>
 17. Kang, B. J., Lee, H. I., Yoon, S. K., Kim, Y. C., Jeong, S. B., O, S. J., Kim, H. (2024). A survey of FPGA and ASIC designs for transformer inference acceleration and optimization. *Journal of Systems Architecture*, 155, 103247. <https://doi.org/10.1016/j.sysarc.2024.103247>
 18. Bjerge, K., Schougaard, J. H., Larsen, D. E. (2021). A scalable and efficient convolutional neural network accelerator using HLS for a system-on-chip design. *Microprocessors and Microsystems*, 87, 104363. <https://doi.org/10.1016/j.micpro.2021.104363>
 19. Yan, Y., Ling, Y., Huang, K., Chen, G. (2023). An efficient real-time accelerator for high-accuracy DNN-based optical flow estimation in FPGA. *Journal of Systems Architecture*, 136, 102818. <https://doi.org/10.1016/j.sysarc.2022.102818>
 20. Tasci, M., Istanbulu, A., Tumen, V., Kosunalp, S. (2025). FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs. *Applied Sciences*, 15 (2), 688. <https://doi.org/10.3390/app15020688>
 21. Liu, Y., Yenamachintala, S. S., Li, P. (2019). Energy-efficient FPGA Spiking Neural Accelerators with Supervised and Unsupervised Spike-timing-dependent-Plasticity. *ACM Journal on Emerging Technologies in Computing Systems*, 15 (3), 1–19. <https://doi.org/10.1145/3313866>
 22. Parashar, A., Raina, P., Shao, Y. S., Chen, Y.-H., Ying, V. A., Mukkara, A. et al. (2019). Timeloop: A Systematic Approach to DNN Accelerator Evaluation. 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 304–315. <https://doi.org/10.1109/ispass.2019.00042>
 23. Chen, Y.-H., Yang, T.-J., Emer, J. S., Sze, V. (2019). Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9 (2), 292–308. <https://doi.org/10.1109/jetcas.2019.2910232>
 24. Jouppi, N. P., Young, C., Patil, N. et al. (2017) In-datacenter performance analysis of a tensor processing unit. *arXiv*. <https://doi.org/10.48550/arXiv.1704.04760>
 25. Dhilleswararao, P., Boppu, S., Manikandan, M. S., Cenkeramaddi, L. R. (2022). Efficient Hardware Architectures for Accelerating Deep Neural Networks: Survey. *IEEE Access*, 10, 131788–131828. <https://doi.org/10.1109/access.2022.3229767>
 26. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., Cong, J. (2015). Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks. *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 161–170. <https://doi.org/10.1145/2684746.2689060>
 27. Chen, T., Moreau, T., Jiang, Z. et al. (2018). TVM: an automated end-to-end optimizing compiler for deep learning. In *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 579–594. Available at: <https://www.usenix.org/system/files/osdi18-chen.pdf>
 28. Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, 105 (12), 2295–2329. <https://doi.org/10.1109/jproc.2017.2761740>
 29. Jouppi, N. P., Hyun Yoon, D., Ashcraft, M., Gottscho, M., Jablin, T. B., Kurian, G. et al. (2021). Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product. 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 1–14. <https://doi.org/10.1109/isca52012.2021.00010>
 30. Mittal, S. (2018). A survey of FPGA-based accelerators for convolutional neural networks. *Neural Computing and Applications*, 32 (4), 1109–1139. <https://doi.org/10.1007/s00521-018-3761-1>

DOI: 10.15587/1729-4061.2026.360407

DEVISING A METHODOLOGY TO COMPARE METHODS FOR SELECTING THE OPTIMAL ARCHITECTURAL VARIATION OF COMMAND QUERY RESPONSIBILITY SEGREGATION WITH EVENT SOURCING (p. 24–44)

Dmytro Hruzin

Oles Honchar Dnipro National University, Dnipro, Ukraine
ORCID: <https://orcid.org/0009-0004-8534-2559>

Oleksandr Lytvynov

Oles Honchar Dnipro National University, Dnipro, Ukraine
ORCID: <https://orcid.org/0000-0001-7660-1353>

This study compares methods for evaluating software architecture. The work is aimed at improving the objectivity of such a comparison by using quantitative indicators obtained on the basis of a parametric model whose adequacy has been verified experimentally.

This paper proposes a methodology to quantitatively compare methods for evaluating software architecture. This methodology enabled a comparison between the architectural trade-off analysis method (ATAM) and decision support technology for architectural variations in the Command Query Responsibility Segregation (CQRS) architecture with the event sourcing approach (DSAV-CQRSES).

The methodology's feature is a parametric model built on the basis of an analysis of real projects, which makes it possible to substantiate the correctness of reference variation, as well as quantitative evaluation of the methods, in particular by the duration of application and the discrepancy of results.

The proposed methodology has made it possible to quantitatively confirm the advantages of DSAV-CQRSES over ATAM for solving the task of choosing the optimal variation of CQRSES architecture.

The duration of ATAM application was 32 hours with the participation of four specialists (total costs of 88 person-hours). The DSAV-CQRSES technology was used by one specialist for 40 hours. With the average and limit parameters of the model, DSAV-CQRSES robustly determined the recommended variation at a variance coefficient of 2.8%. In contrast, ATAM demonstrated lower accuracy and provided the correct choice in three out of five cases with a variation coefficient of 33.64%.

The results of the experiment could be used in practice to select tools for comparing variations in the CQRSES architecture.

Keywords: optimization, computational experiment, ATAM, DSAV-CQRSES, data analysis, comparative analysis.

References

1. Betts, D., Dominguez, J., Melnik, G., Simonazzi, F., Subramanian, M. (2013). Exploring CQRS and Event Sourcing: A Journey into High Scalability, Availability, and Maintainability with Windows Azure. *Microsoft patterns & practices*, 376. Available at: https://download.microsoft.com/download/e/a/8/ea8c6e1f-01d8-43ba-992b-35cfcaa4fae3/cqrs_journey_guide.pdf

2. CQRS Documents by Greg Young. Available at: https://cQRS.files.wordpress.com/2010/11/cQRS_documents.pdf
3. Breivold, H. P., Crnkovic, I., Larsson, M. (2012). A systematic review of software architecture evolution research. *Information and Software Technology*, 54 (1), 16–40. <https://doi.org/10.1016/j.infsof.2011.06.002>
4. Ford, N., Parsons, R., Kua, P., Sadalage, P. (2022). *Building Evolutionary Architectures*. O'Reilly Media, 262. Available at: <https://www.oreilly.com/library/view/building-evolutionary-architectures/9781492097532/>
5. Soliman, M., Avgeriou, P., Li, Y. (2021). Architectural design decisions that incur technical debt – An industrial case study. *Information and Software Technology*, 139, 106669. <https://doi.org/10.1016/j.infsof.2021.106669>
6. Sobhy, D., Bahsoon, R., Minku, L., Kazman, R. (2021). Evaluation of Software Architectures under Uncertainty. *ACM Transactions on Software Engineering and Methodology*, 30 (4), 1–50. <https://doi.org/10.1145/3464305>
7. Babar, M. A., Zhu, L., Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods. 2004 Australian Software Engineering Conference. Proceedings, 309–318. <https://doi.org/10.1109/aswec.2004.1290484>
8. Roy, B., Graham, N. (2008). *Methods for Evaluating Software Architecture: A Survey*. Technical Report No. 2008-545. School of Computing Queen's University at Kingston Ontario. Available at: https://research.cs.queensu.ca/TechReports/Reports/2008-545.pdf?utm_source=chatgpt.com
9. Mattsson, M., Grahn, H., Mårtensson, F. (2006). *Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability*. Presented at the Second International Conference on the Quality of Software Architectures (QoSA 2006). Västerås. Available at: https://www.researchgate.net/publication/200622064_Software_Architecture_Evaluation_Methods_for_Performance_Maintainability_Testability_and_Portability
10. Kazman, R., Klein, M., Clements, P. (2000). *ATAM: Method for Architecture Evaluation*. Technical report: CMU/SEI-2000-TR-004, ESC-TR-2000-004. Pittsburgh: Product Line Systems, 71. Available at: https://www.researchgate.net/publication/2814191_ATAM_Method_for_Architecture_Evaluation
11. Lytvynov, O., Hruzin, D. (2025). Decision-making on Command Query Responsibility Segregation with Event Sourcing architectural variations. *Technology Audit and Production Reserves*, 4 (2 (84)), 37–59. <https://doi.org/10.15587/2706-5448.2025.337168>
12. Abrahão, S., Insfran, E. (2017). Evaluating Software Architecture Evaluation Methods. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 144–153. <https://doi.org/10.1145/3084226.3084253>
13. Gonzalez-Huerta, J., Insfran, E., Abrahão, S., Scanniello, G. (2015). Validating a model-driven software architecture evaluation and improvement method: A family of experiments. *Information and Software Technology*, 57, 405–429. <https://doi.org/10.1016/j.infsof.2014.05.018>
14. Eloranta, V.-P., van Heesch, U., Avgeriou, P., Harrison, N., Koskimies, K. (2014). Lightweight Evaluation of Software Architecture Decisions. *Relating System Quality and Software Architecture*, 157–179. <https://doi.org/10.1016/b978-0-12-417009-4.00006-5>
15. Babar, M. A., Kitchenham, B. (2007). Assessment of a Framework for Comparing Software Architecture Analysis Methods. *Electronic Workshops in Computing*. <https://doi.org/10.14236/ewic/ease2007.2>
16. Babar, M. A., Gorton, I. (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. 11th Asia-Pacific Software Engineering Conference, 600–607. <https://doi.org/10.1109/apsec.2004.38>
17. Hruzin, D. L., Lytvynov, O. A. (2025). The impact of quality attribute selection on comparing CQRS with event sourcing architectural variations. *Proceeding of the XVIII International Scientific and Practical Conference*. Odesa, 614–616. Available at: <https://if.uu.edu.ua/wp-content/uploads/2025/11/zbirnyk-tez-konferentsii-itia-2025.pdf>
18. Le Thi Kim Nhung, H., Van Hai, V., Hoc, H. T. (2021). Analyzing Correlation of the Relationship Between Technical Complexity Factors and Environmental Complexity Factors for Software Development Effort Estimation. *Software Engineering Application in Informatics*, 835–848. https://doi.org/10.1007/978-3-030-90318-3_65
19. da Silva, A. R., Savić, D., Vlajić, S., Antović, I., Lazarević, S., Stanojević, V., Milić, M. (2015). A pattern language for use cases specification. *Proceedings of the 20th European Conference on Pattern Languages of Programs*, 1–18. <https://doi.org/10.1145/2855321.2855330>
20. Minsky, M. (1974). *A Framework for Representing Knowledge*. MIT Research Lab Technical Report. Massachusetts Institute of Technology, Cambridge, MA, United States. Available at: <https://courses.media.mit.edu/2004spring/mas966/Minsky%201974%20Framework%20for%20knowledge.pdf>
21. Hruzin, D. (2026). *Classification of Use Cases in Systems Based on CQRS with Event Sourcing*. Zenodo. <https://doi.org/10.5281/zenodo.18621093>
22. Janssen, P., Stouffs, R. (2015). Types of Parametric Modelling. *Proceedings of the 20th Conference on Computer Aided Architectural Design Research in Asia (CAADRIA)*, 157–166. <https://doi.org/10.52842/conf.caadria.2015.157>
23. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29 (5). <https://doi.org/10.1214/aos/1013203451>
24. Kohl, M., Ruckdeschel, P. (2010). R Package distrMod: S4 Classes and Methods for Probability Models. *Journal of Statistical Software*, 35 (10). <https://doi.org/10.18637/jss.v035.i10>
25. General Data Protection Regulation. Available at: <https://gdpr-info.eu/>
26. Aljournah, E., Al-Mousawi, F., Ahmad, I., Al-Shammri, M., Al-Jady, Z. (2015). SLA in Cloud Computing Architectures: A Comprehensive Study. *International Journal of Grid and Distributed Computing*, 8 (5), 7–32. <https://doi.org/10.14257/ijgdc.2015.8.5.02>
27. *Application Performance Index – Apdex Final Technical Specification (2005)*. Apdex Alliance. Available at: https://www.apdex.org/wp-content/uploads/2020/09/Apdex_Technical_Specification.pdf
28. Tsyganok, I. (2016). Pair-Programming from a Beginner's Perspective. *Agile Processes, in Software Engineering, and Extreme Programming*, 270–277. https://doi.org/10.1007/978-3-319-33515-5_25
29. Hruzin, D. (2026). *dmytrohruzin/DSAV-CQRSES-ATAM-parametric-model: TODOs for MVP (1.0.2)*. Zenodo. <https://doi.org/10.5281/zenodo.18473292>
30. *DSAV-CQRSES-ATAM-parametric-model*. Available at: https://sonarcloud.io/summary/overall?id=dmytrohruzin_DSAV-CQRSES-ATAM-parametric-model&branch=main
31. CQRS pattern. Available at: <https://learn.microsoft.com/en-us/azure/architecture/patterns/cQRS>
32. EventSourcingpattern. Available at: <https://learn.microsoft.com/en-us/azure/architecture/patterns/event-sourcing>

33. Fowler, M. (2011). CQRS. Available at: <https://martinfowler.com/bliki/CQRS.html>
34. Di Pompeo, D., Tucci, M. (2023). Quality Attributes Optimization of Software Architecture: Research Challenges and Directions. 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), 252–255. <https://doi.org/10.1109/icsa-c57050.2023.00061>
35. Hruzyn, D. (2026). Presentation: Choose the optimal variation of CQRS with ES architecture with ATAM. <https://doi.org/10.5281/zenodo.18472924>
36. Taguchi, G., Chowdhury, S., Wu, Y. (2004). Taguchi's Quality Engineering Handbook. Wiley. <https://doi.org/10.1002/9780470258354>
37. Lytvynov, O., Khandetskiy, V., Lytvynov, M. (2025). The Estimation of Effort for Domain-Driven Architectural Variations Migration using Modified Use Case Size Points Method. Proceeding of the 1st International Scientific and Practical Conference (MIT@AIS'2025s). Kharkiv-Yaremche, 149–152. Available at: <https://drive.usercontent.google.com/u/0/uc?id=1iVEiy8u0IJ9jgcQyvyhRiId3vKpAyGf>
38. Hruzyn, D. (2026). Calculation of cognitive functional complexity of activities for processes in the CQRS with Event sourcing systems. Zenodo. <https://doi.org/10.5281/zenodo.18509043>
39. Wang, Y., Shao, J. (2003). Measurement of the cognitive functional complexity of software. The Second IEEE International Conference on Cognitive Informatics, 2003. Proceedings, 67–74. <https://doi.org/10.1109/coginf.2003.1225955>
40. Ionita, M. T., America, P., Hammer, D. K., Obbink, H., Trienekens, J. J. M. A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), 277–280. <https://doi.org/10.1109/wicsa.2004.1310709>
41. Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8 (3), 338–353. [https://doi.org/10.1016/s0019-9958\(65\)90241-x](https://doi.org/10.1016/s0019-9958(65)90241-x)
42. Brunelli, M. (2015). Introduction to the Analytic Hierarchy Process. In (Editor), SpringerBriefs in Operations Research. Springer International Publishing. <https://doi.org/10.1007/978-3-319-12502-2>
43. DBBSoftware. Available at: <https://dbbsoftware.com/>

DOI: 10.15587/1729-4061.2026.362735

DEVISING A CROSS-PROJECT LEARNING METHOD FOR SOFTWARE DEFECT PREDICTION BASED ON FUZZY ALGORITHMIC MODELS (p. 45–55)

Hanna Rakytyanska

Vinnitsia National Technical University, Vinnitsia, Ukraine
ORCID: <https://orcid.org/0000-0001-5863-3730>

Bohdan Prus

Vinnitsia National Technical University, Vinnitsia, Ukraine
ORCID: <https://orcid.org/0009-0008-7214-0949>

This work investigates fuzzy algorithmic models for predicting software defects. An issue related to conventional machine (deep) learning models is complexity and interpretability. In addition, cross-project learning of such models requires solving the problem of heterogeneity in the initial and target data distributions.

A fuzzy algorithmic model for predicting software defects with an integrated distribution of implementation options for development stages has been proposed. A genetic-neural method for tuning a fuzzy algorithmic model to cross-project data has been devised.

Unlike machine learning models, the interpretability of the prediction model was achieved by assessing the correctness (defectivity)

of development stages using fuzzy rules. The model is built on the basis of “work-control-refinement” algorithmic structures, which is an analog of a fuzzy knowledge base. Fuzzy estimates of the defectivity of the execution of work, control, and refinement operators are subject to tuning, where defect ranks model the distribution of resources. The training sample is formed on the basis of estimates of the correctness of implementation options for algorithmic structures.

The integration of indicators for implementation options for work, control, and refinement operators has made it possible to solve the issue of heterogeneity in the initial and target data distributions. Unlike known methods for training neural-fuzzy models of software reliability, simplification of the tuning process was achieved by transferring cross-project data.

The scope of practical application includes predicting the quality of new software based on the experience of completed projects. The condition of use is a discrete algorithmic model of the development process.

Keywords: software defect prediction, fuzzy algorithmic model, cross-project transfer learning.

References

1. Malhotra, R., Chawla, S., Sharma, A. (2023). Software defect prediction using hybrid techniques: a systematic literature review. *Soft Computing*, 27 (12), 8255–8288. <https://doi.org/10.1007/s00500-022-07738-w>
2. Pal, S., Sillitti, A. (2022). Cross-Project Defect Prediction: A Literature Review. *IEEE Access*, 10, 118697–118717. <https://doi.org/10.1109/access.2022.3221184>
3. Rotshtein, O. P., Shtovba, S. D., Kozachko, O. M. (2007). Modelivannia ta optymizatsiya nadiynosti bahatovymirnykh alhorytmichnykh protsesiv. *Vinnytsia: «UNIVERSUM»*, 211. Available at: <http://ir.lib.vntu.edu.ua/handle/123456789/2296>
4. Rotshtein, A. P., Rakytyanska, H. B. (2012). Fuzzy Evidence in Identification, Forecasting and Diagnosis. *Studies in Fuzziness and Soft Computing*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-25786-5>
5. Rakytyanska, H., Prus, B. (2025). Fuzzy-algorithmic analysis of software reliability. *Information Technology and Computer Engineering*, 22, 136–147. <https://doi.org/10.31649/vitce/3.2025.136>
6. Rakytyanska, H. (2023). Knowledge Distillation in Granular Fuzzy Models by Solving Fuzzy Relation Equations. *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems*, 95–133. https://doi.org/10.1007/978-3-031-32095-8_4
7. Singh, M., Chhabra, J. K. (2024). Machine learning based improved cross-project software defect prediction using new structural features in object oriented software. *Applied Soft Computing*, 165, 112082. <https://doi.org/10.1016/j.asoc.2024.112082>
8. Abbas, S., Aftab, S., Adnan Khan, M., M. Ghazal, T., Al Hamadi, H., Yeob Yeun, C. (2023). Data and Ensemble Machine Learning Fusion Based Intelligent Software Defect Prediction System. *Computers, Materials & Continua*, 75 (3), 6083–6100. <https://doi.org/10.32604/cmc.2023.037933>
9. Sharma, T., Jatain, A., Bhaskar, S., Pabreja, K. (2023). Ensemble Machine Learning Paradigms in Software Defect Prediction. *Procedia Computer Science*, 218, 199–209. <https://doi.org/10.1016/j.procs.2023.01.002>
10. Dong, X., Wang, J., Liang, Y. (2025). A Novel Ensemble Classifier Selection Method for Software Defect Prediction. *IEEE Access*, 13, 25578–25597. <https://doi.org/10.1109/access.2025.3537658>

11. Omer, A., Rathore, S. S., Kumar, S. (2024). ME-SFP: A Mixture-of-Experts-Based Approach for Software Fault Prediction. *IEEE Transactions on Reliability*, 73 (1), 710–725. <https://doi.org/10.1109/tr.2023.3295012>
12. Wang, S., Huang, L., Gao, A., Ge, J., Zhang, T., Feng, H. et al. (2023). Machine/Deep Learning for Software Engineering: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 49 (3), 1188–1231. <https://doi.org/10.1109/tse.2022.3173346>
13. Goyal, S. R. (2026). Effective software defect prediction with deep neural networks. *Results in Engineering*, 29, 108378. <https://doi.org/10.1016/j.rineng.2025.108378>
14. Sun, T., Allix, K., Kim, K., Zhou, X., Kim, D., Lo, D. et al. (2023). DexBERT: Effective, Task-Agnostic and Fine-Grained Representation Learning of Android Bytecode. *IEEE Transactions on Software Engineering*, 49 (10), 4691–4706. <https://doi.org/10.1109/tse.2023.3310874>
15. Villoth, J. P., Zivkovic, M., Zivkovic, T., Abdel-salam, M., Hammad, M., Jovanovic, L. et al. (2025). Two-tier deep and machine learning approach optimized by adaptive multi-population firefly algorithm for software defects prediction. *Neurocomputing*, 630, 129695. <https://doi.org/10.1016/j.neucom.2025.129695>
16. Abdu, A., Zhai, Z., Abdo, H. A., Algabri, R. (2024). Software Defect Prediction Based on Deep Representation Learning of Source Code From Contextual Syntax and Semantic Graph. *IEEE Transactions on Reliability*, 73 (2), 820–834. <https://doi.org/10.1109/tr.2024.3354965>
17. Bal, P. R., Kumar, S. (2025). An Approach for Cross Project Defect Prediction Using Identical Metrics Matching and Deep Neural Network. *IEEE Transactions on Reliability*, 74 (2), 2678–2692. <https://doi.org/10.1109/tr.2024.3435709>
18. Omondiagbe, O. P., Licorish, S. A., MacDonell, S.G. (2024). Improving transfer learning for software cross-project defect prediction. *Applied Intelligence*, 54, 5593–5616. <https://doi.org/10.1007/s10489-024-05459-1>
19. Nevendra, M., Singh, P. (2025). TRGNet: a deep transfer learning approach for software defect prediction. *Expert Systems with Applications*, 282, 127799. <https://doi.org/10.1016/j.eswa.2025.127799>
20. Tong, H., Zhang, D., Liu, J., Xing, W., Lu, L., Lu, W., Wu, Y. (2024). MASTER: Multi-Source Transfer Weighted Ensemble Learning for Multiple Sources Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*, 50 (5), 1281–1305. <https://doi.org/10.1109/tse.2024.3381235>
21. Elsabagh, M. A., Emam, O. E., Gafar, M. G., Medhat, T. (2023). Handling uncertainty issue in software defect prediction utilizing a hybrid of ANFIS and turbulent flow of water optimization algorithm. *Neural Computing and Applications*, 36 (9), 4583–4602. <https://doi.org/10.1007/s00521-023-09315-0>
22. Lakshmi, P. J., Krishna, T. V. S., Kumar, N. B., Rao, D. S. N. M., Hosseinpour, A., Lenin, N. C. (2025). Improving Software Fault Prediction With a Hybrid DE-WOA Optimizer and ANFIS-Enhanced Ensemble Learning. *IEEE Access*, 13, 158194–158209. <https://doi.org/10.1109/access.2025.3603980>
23. Azzeh, M., Elsheikh, Y., Alqasrawi, Y. (2024). Software defect density prediction using grey system theory and fuzzy logic. *Soft Computing*, 28 (21-22), 12897–12916. <https://doi.org/10.1007/s00500-024-10324-x>
24. Azzeh, M., Abdel-Rahman, M. J. (2026). Cross-project software defects prediction using fuzzy embedding and deep learning. *Information and Software Technology*, 190, 107968. <https://doi.org/10.1016/j.infsof.2025.107968>
25. Behera, A. K., Panda, M., Dehuri, S. (2024). A recurrent ANFIS tuned by modified differential evolution for efficient prediction of software reliability. *Evolutionary Intelligence*, 17 (5-6), 3469–3482. <https://doi.org/10.1007/s12065-024-00940-9>
26. Chatterjee, S., Saha, D. (2024). IT2F-SEDNN: an interval type-2 fuzzy logic-based stacked ensemble deep learning approach for early phase software dependability analysis. *Innovations in Systems and Software Engineering*, 21 (2), 727–746. <https://doi.org/10.1007/s11334-024-00563-4>
27. Rakytyanska, H. (2015). Fuzzy classification knowledge base construction based on trend rules and inverse inference. *Eastern-European Journal of Enterprise Technologies*, 1 (3 (73)), 25–32. <https://doi.org/10.15587/1729-4061.2015.36934>
28. Rakytyanska, H. (2017). Optimization of fuzzy classification knowledge bases using improving transformations. *Eastern-European Journal of Enterprise Technologies*, 5 (2 (89)), 33–41. <https://doi.org/10.15587/1729-4061.2017.110261>
29. Rakytyanska, H., Prus, B. (2024). Constructing Prototype-Based Granular Fuzzy Rules for Scene Classification on Mobile Devices. Vol. 1. *Lecture Notes in Data Engineering, Computational Intelligence, and Decision-Making*, 194–218. https://doi.org/10.1007/978-3-031-70959-3_10

DOI: 10.15587/1729-4061.2026.360693

DEVELOPMENT OF A METHOD FOR AUTOMATED INTEGRATION OF THE “PENTAGON” TESTING MODEL INTO A CONTINUOUS INTEGRATION AND DELIVERY PIPELINE FOR MICROSERVICE SOFTWARE (p. 56–66)

Oleh Kuzmych

Lviv Polytechnic National University, Lviv, Ukraine
ORCID: <https://orcid.org/0000-0002-9749-6385>

Maksym Seniv

Lviv Polytechnic National University, Lviv, Ukraine
ORCID: <https://orcid.org/0000-0003-1044-4628>

This study investigates the process of verifying the reliability of software with a microservice architecture, developed by several teams in parallel. The task addressed relates to the lack of cost-effective methods for automated integration of testing models into continuous integration and delivery (CI/CD) pipelines adapted to the microservice architecture. This leads to the emergence of “blind spots” in the field of functional and non-functional requirements and excessive time consumption of developers.

This paper reports a method devised for automated integration of the Pentagon testing model into a multi-stage continuous integration and delivery pipeline. The method implements the cumulative principle of progressive distribution of six testing layers between four isolated environments: DEV (development), TEST (test), PRE-PROD (pre-production), and PROD (production). The devised method guarantees coverage of all six layers during each change in the code base.

The key feature of the method is to eliminate the influence of human factor on decisions to launch tests. This makes it possible to achieve increased reliability values, namely Defect Removal Efficiency (DRE), while the manual method detects on average only 27% of defects. The devised method reduces the time when the developer is involved in the process of running tests by 70% compared to the manual approach. For 100 commits (fixing the current state of authentic code and files in the local repository) per month, the savings compared to the manual method are 37%.

The method is suitable for verifying software with a microservice architecture, which is deployed on cloud platforms (Amazon Web

Services, USA; Azure, USA; Google Cloud, USA) with support for automated testing and container orchestration.

Keywords: microservice architecture, continuous integration and delivery pipeline, software reliability.

References

- Humble, J., Forsgren, N., Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. Portland: IT Revolution Press, 286. Available at: <https://ebooks.karbust.me/Technology/Accelerate%20The%20Science%20of%20Lean%20Software%20and%20DevOps%20Building%20and%20Scaling%20High%20Performing%20Technology%20Organizations%20by%20Nicole%20Forsgren%20Jez%20Humble%20Gene%20Kim.pdf>
- Boehm, B., Basili, V. R. (2001). Top 10 list [software development]. *Computer*, 34 (1), 135–137. <https://doi.org/10.1109/2.962984>
- Ponce, F., Verdecchia, R., Miranda, B., Soldani, J. (2025). Microservices testing: A systematic literature review. *Information and Software Technology*, 188, 107870. <https://doi.org/10.1016/j.infsof.2025.107870>
- Waseem, M., Liang, P., Marquez, G., Salle, A. D. (2020). Testing Microservices Architecture-Based Applications: A Systematic Mapping Study. 2020 27th Asia-Pacific Software Engineering Conference (APSEC), 119–128. <https://doi.org/10.1109/apsec51365.2020.00020>
- Wang, Y., Mäntylä, M. V., Liu, Z., Markkula, J. (2022). Test automation maturity improves product quality – Quantitative study of open source projects using continuous integration. *Journal of Systems and Software*, 188, 111259. <https://doi.org/10.1016/j.jss.2022.111259>
- Laukkanen, E., Itkonen, J., Lassenius, C. (2017). Problems, causes and solutions when adopting continuous delivery – A systematic literature review. *Information and Software Technology*, 82, 55–79. <https://doi.org/10.1016/j.infsof.2016.10.001>
- Shahin, M., Ali Babar, M., Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5, 3909–3943. <https://doi.org/10.1109/access.2017.2685629>
- Mascheroni, M. A., Irrazábal, E. (2018). Continuous Testing and Solutions for Testing Problems in Continuous Delivery: A Systematic Literature Review. *Computación Y Sistemas*, 22 (3). <https://doi.org/10.13053/cys-22-3-2794>
- Zhang, Y., Vasilescu, B., Wang, H., Filkov, V. (2018). One size does not fit all: an empirical study of containerized continuous deployment workflows. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 295–306. <https://doi.org/10.1145/3236024.3236033>
- Zhou, X., Peng, X., Xie, T., Sun, J., Ji, C., Li, W., Ding, D. (2021). Fault Analysis and Debugging of Microservice Systems: Industrial Survey, Benchmark System, and Empirical Study. *IEEE Transactions on Software Engineering*, 47 (2), 243–260. <https://doi.org/10.1109/tse.2018.2887384>
- Jernberg, H., Runeson, P., Engström, E. (2020). Getting Started with Chaos Engineering - design of an implementation framework in practice. *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–10. <https://doi.org/10.1145/3382494.3421464>
- Basiri, A., Behnam, N., de Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., Rosenthal, C. (2016). *Chaos Engineering*. *IEEE Software*, 33 (3), 35–41. <https://doi.org/10.1109/ms.2016.60>
- Heorhiadi, V., Rajagopalan, S., Jamjoom, H., Reiter, M. K., Sekar, V. (2016). Gremlin: Systematic Resilience Testing of Microservices. 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 57–66. <https://doi.org/10.1109/icdcs.2016.11>
- Meiklejohn, C. S., Estrada, A., Song, Y., Miller, H., Padhye, R. (2021). Service-Level Fault Injection Testing. *Proceedings of the ACM Symposium on Cloud Computing*, 388–402. <https://doi.org/10.1145/3472883.3487005>
- Bouizem, Y., Dib, D., Parlavantzas, N., Morin, C. (2023). Integrating request replication into FaaS platforms: an experimental evaluation. *Journal of Cloud Computing*, 12 (1). <https://doi.org/10.1186/s13677-023-00457-z>
- Kuzmich, O., Seniv, M. (2024). An Improved Approach to Increase the Fault Tolerance of Microservice Software Through Automated Functional and Fault Tolerance Testing. 2024 IEEE 19th International Conference on Computer Science and Information Technologies (CSIT), 1–4. <https://doi.org/10.1109/csit65290.2024.10982594>
- Cohn, M. (2009). *Succeeding with Agile: Software Development Using Scrum*. Boston: Addison-Wesley Professional, 504. Available at: https://www.researchgate.net/publication/234803335_Succeeding_with_Agile_Software_Development_Using_Scrum
- Welcome to the State of Developer Ecosystem Report 2024. *JetBrains*. Available at: <https://www.jetbrains.com/lp/devecosystem-2024/>
- State of Cloud Native Development Q3 2025. *CNCF*. Available at: https://www.cncf.io/wp-content/uploads/2025/11/cncf_report_stateofcloud_111025a.pdf
- The State of Chaos Engineering in 2021. *Gremlin Inc*. Available at: <https://www.gremlin.com/blog/the-state-of-chaos-engineering-in-2021>
- Rubinstein, R. Y., Kroese, D. P. (2016). *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Statistics. <https://doi.org/10.1002/9781118631980>
- AWS Pricing. *Amazon Web Services*. Available at: <https://aws.amazon.com/pricing/>
- Developer Survey 2024. *Stack Overflow*. Available at: <https://survey.stackoverflow.co/2024/>
- The Economic Impacts of Inadequate Infrastructure for Software Testing (Planning Report 02-3). *National Institute of Standards and Technology (NIST)*. Available at: <https://www.nist.gov/system/files/documents/director/planning/report02-3.pdf>
- Yakovyna, V., Seniv, M., Symets, I. (2020). The Relation between Software Development Methodologies and Factors Affecting Software Reliability. 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), 377–381. <https://doi.org/10.1109/csit49958.2020.9321937>

DOI: 10.15587/1729-4061.2026.363077

DEVELOPMENT OF A MATHEMATICAL MODEL FOR SIZE ESTIMATION OF JAVA SPRING WEB APPLICATIONS USING REGRESSION ANALYSIS (p. 67–74)

Lidiia Makarova

Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine

ORCID: <https://orcid.org/0000-0003-2903-3001>

Liudmyla Latanska

Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine

ORCID: <https://orcid.org/0000-0001-6473-7624>

Andrii PukhalevychAdmiral Makarov National University of Shipbuilding,
Mykolaiv, UkraineORCID: <https://orcid.org/0000-0002-8827-3251>**Vladimir Kairov**Admiral Makarov National University of Shipbuilding,
Mykolaiv, UkraineORCID: <https://orcid.org/0000-0003-0502-7942>**Maksym Dzhuryynksyi**Admiral Makarov National University of Shipbuilding,
Mykolaiv, UkraineORCID: <https://orcid.org/0009-0002-0685-9038>

This study investigates the process to estimate the size of web applications developed in Java (country of origin – USA) using the Spring framework (country of origin – USA).

The task addressed is to improve reliability in estimating the size of the corresponding web applications. Estimating the size of web applications developed in Java using the Spring framework is an important task in software engineering. This makes it possible to assess the required time frame for implementing the functionality and to plan a project budget.

As a result of the work, a mathematical model was built for estimating the size of web applications in Java and Spring based on the normalizing transformation of the decimal logarithm; a corresponding program was developed to automate calculations. Quality parameters of the constructed model are as follows: $R^2 = 0.9173$, $MMRE = 0.1511$, $PRED(0.25) = 0.7931$.

That has made it possible to improve the reliability of estimating the size of such web applications, namely, to reduce the value of average relative error and increase the level of prediction, as well as to narrow the widths of the confidence and prediction intervals.

A data set consisting of 36 projects was collected; its preprocessing was performed, which included checking for multivariate normality of the distribution and normalization by logarithmization. An appropriate nonlinear regression model was constructed. To improve the reliability of the model, an algorithm was applied that includes iterative removal of outliers using the square of the Mahalanobis distance and Fisher's criterion. A program was developed based on the constructed nonlinear regression model; the results were analyzed.

Analysis of the quality of the constructed model reveals its adequacy and applicability for solving tasks of estimating the size of the corresponding software.

Keywords: size estimation, normalization, nonlinear regression, outlier, Java, Spring, software.

References

1. TIOBE Index. Available at: <https://www.tiobe.com/tiobe-index/>
2. Tan, H. B. K., Zhao, Y., Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. *ACM Transactions on Software Engineering and Methodology*, 19 (2), 1–37. <https://doi.org/10.1145/1571629.1571630>
3. Daud, M., Afzal Malik, A. (2022). Construction and Validation of Early Software Size Estimation Models Based on ADAF-Adjusted ACD Metrics. *The Computer Journal*, 66 (9), 2123–2137. <https://doi.org/10.1093/comjnl/bxac065>
4. Daud, M., Malik, A. A. (2024). Exploring the Impact of Security-Based Non-Functional Requirements on Early Software Size Estimation. *Computing and Informatics*, 43 (5), 1234–1255. https://doi.org/10.31577/cai_2024_5_1234
5. Chatterjee, S., Hadi, A. S. (2006). *Regression Analysis by Example*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc. <https://doi.org/10.1002/0470055464>
6. Oriekhov, O., Farionova, T., Chernova, L., Chernova, L., Vorona, M. (2024). Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications. *CEUR Workshop Proceedings*, 3709, 54–66. Available at: <https://ceur-ws.org/Vol-3709/paper5.pdf>
7. Prykhodko, S. B., Prykhodko, N. V., Koltsov, A. V. (2024). A Nonlinear Regression Model for Early Loc Estimation of Open-Source Kotlin-Based Applications. *Radio Electronics, Computer Science, Control*, 1, 85. <https://doi.org/10.15588/1607-3274-2024-1-8>
8. Prykhodko, S., Prykhodko, A., Shutko, I. (2021). Estimating the Size of Web Apps Created Using the CakePHP Framework by Nonlinear Regression Models with Three Predictors. *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, 333–336. <https://doi.org/10.1109/csit52700.2021.9648680>
9. Prykhodko, S. B., Prykhodko, N. V. (2021). A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations. *CEUR Workshop Proceedings*, 3179, 156–166. Available at: https://ceur-ws.org/Vol-3179/Paper_15.pdf
10. Makarova, L. M., Prykhodko, N. V., Kudin, O. O. (2019). Constructing the non-linear regression model for size estimation of web-applications implemented in Java. *Visnyk KhNTU*, 2 (69), 145–153. Available at: <https://journals.kntu.net.ua/index.php/visnyk/article/view/417>
11. Aniche, M. Java code metrics calculator (CK). GitHub. Available at: <https://github.com/mauricioaniche/ck>
12. Java. Oracle. Available at: <https://www.oracle.com/java/technologies/>
13. IntelliJ IDEA. The Leading IDE for Professional Development in Java and Kotlin. Available at: <https://www.jetbrains.com/idea/>
14. Alasadi, S. A., Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12 (16), 4102–4107. Available at: https://www.researchgate.net/publication/319990923_Review_of_Data_Preprocessing_Techniques_in_Data_Mining
15. Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57 (3), 519–530. <https://doi.org/10.1093/biomet/57.3.519>
16. Mardia, K. V., Kent, J. T., Taylor, C. C. (2024). *Multivariate Analysis*. John Wiley & Sons, 592. Available at: <https://www.wiley.com/Multivariate+Analysis%2C+2nd+Edition-p-9781118738023>
17. Prykhodko, S., Prykhodko, N., Makarova, L. (2018). Estimating the software size of open-source PHP-based systems using non-linear regression analysis. *CEUR Workshop Proceedings*, 2300, 199–202. Available at: <https://ceur-ws.org/Vol-2300/Paper48.pdf>
18. Prykhodko, S., Prykhodko, N., Makarova, L., Pukhalevych, A. (2018). Application of the squared mahalanobis distance for detecting outliers in multivariate non-Gaussian data. *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 962–965. <https://doi.org/10.1109/tcset.2018.8336353>
19. Larose, D. T., Larose, C. D. (2015). *Data Mining and Predictive Analytics*. John Wiley & Sons, 824. Available at: https://lmsspada.kemdiktisaintek.go.id/pluginfile.php/749864/mod_resource/content/4/Data%20Mining%20and%20Predictive%20Analytics.pdf
20. GitHub. Available at: <https://github.com/>
21. Oriekhov, O., Farionova, T., Chernova, L. (2024). Three-factor nonlinear regression model of estimating the size of Java-software. *CEUR Work-*

shop Proceedings, 3790, 506–518. Available at: <https://ceur-ws.org/Vol-3790/paper44.pdf>

DOI: 10.15587/1729-4061.2026.363741

DEVELOPMENT OF A TWO-LAYER AGENTIC ARCHITECTURE FOR AUTOMATED GENERATION AND VALIDATION OF WAGON USER INTERFACE SCHEMAS USING MULTIMODAL LARGE LANGUAGE MODELS (p. 75–86)

Ivan Dobrovolskyi

Independent Researcher, San Jose, California, USA

ORCID: <https://orcid.org/0009-0005-6938-0384>

The object of the study is automated generation and validation of wagon interface schemas in railway ticket booking systems. Manual updating of one configuration takes approximately 24 hours. It depends on software releases, requires engineer involvement, and increases the risk of production errors.

A two-layer agentic architecture is proposed. It includes separation of reasoning and execution, multimodal conversion of schematics into structured configurations, and deterministic validation. The architecture was tested on the Ukrzaliznytsia platform. The platform supports more than 150 wagon configurations and serves over 20 million passengers per year.

Twenty production schematics of different wagon classes were processed. One schematic required manual correction. The others were refined through one or two additional prompts. The time required to create a new wagon type decreased from approximately 24 hours to 15 minutes. This corresponds to an approximately 99% reduction. Monthly engineering involvement was largely replaced by a serverless workflow with infrastructure costs of less than 5 USD per month. No incidents caused by model hallucinations were recorded.

The solution supports more than 150 configurations without loading all schemas into a single context. It enables non-technical administrators to update configurations outside release cycles and reduces dependence on engineering teams. The approach integrates mandatory pre-deployment validation into the configuration creation process. Structure, format, and correctness remain subject to formal control. This reduces manual engineering work without changes to the core application code. The approach is applicable to railway ticketing and related systems where physical objects are represented as digital interface schemas.

Keywords: ticketing automation, reasoning-execution separation, production validation, computer vision, serverless deployment.

References

- Dpty PM Kuleba announces record transportation of Ukrzaliznytsia in 2024. Interfax-Ukraine. Available at: <https://en.interfax.com.ua/news/economic/1034288.html>
- According to the results of 2024, 86% of railway tickets were purchased online (2025). Ukrzaliznytsia. Available at: <https://t.me/UkrzalInfo/6461>
- Berrios Villalba, A. (2020). How to Speed Up Digitization in the Railway [Viewpoint]. IEEE Electrification Magazine, 8 (1), 76–75. <https://doi.org/10.1109/mele.2019.2962895>
- Cecchetti, G., Lina, A., Ruscelli, Uliyanov, C., Hyde, P., Liu, J. et al. (2023). Toward new generation railway Traffic Management Systems: the contribution of the OPTIMA project. Transportation Research Procedia, 72, 3166–3173. <https://doi.org/10.1016/j.trpro.2023.11.882>
- Bezuidenhout, M., Jooste, J. L., Lucke, D., Fourie, C. J. (2023). Leveraging digitilisation and machine learning for improved railway operations and maintenance. Procedia CIRP, 120, 702–707. <https://doi.org/10.1016/j.procir.2023.09.062>
- Jiang, J., Wang, F., Shen, J., Kim, S., Kim, S. (2026). A Survey on Large Language Models for Code Generation. ACM Transactions on Software Engineering and Methodology, 35 (2), 1–72. <https://doi.org/10.1145/3747588>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J. et al. (2021). Evaluating large language models trained on code. arXiv. <https://doi.org/10.48550/arXiv.2107.03374>
- Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J. et al. (2023). Gemini: a family of highly capable multimodal models. arXiv. <https://doi.org/10.48550/arXiv.2312.11805>
- Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G. et al. (2024). Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. arXiv. <https://doi.org/10.48550/arXiv.2403.05530>
- Pasichnyk, V., Horlatch, V. (2025). Automated extraction of key parameters and detection of inconsistencies in clinical documentation using large language models. Eastern-European Journal of Enterprise Technologies, 6 (2 (138)), 6–14. <https://doi.org/10.15587/1729-4061.2025.337915>
- Zhuravchak, A., Piskozub, A., Skorynovych, B., Lakh, Y., Zhuravchak, D., Hlushchenko, P. et al. (2025). Design and development of a large language model-based tool for vulnerability detection. Eastern-European Journal of Enterprise Technologies, 2 (2 (134)), 75–83. <https://doi.org/10.15587/1729-4061.2025.325251>
- Sarp, S., Kuzlu, M., Jovanovic, V., Polat, Z., Guler, O. (2024). Digitalization of railway transportation through AI-powered services: digital twin trains. European Transport Research Review, 16 (1). <https://doi.org/10.1186/s12544-024-00679-5>
- De Donato, L., Dirnfeld, R., Somma, A., De Benedictis, A., Flammini, F., Marrone, S. et al. (2023). Towards AI-assisted digital twins for smart railways: preliminary guideline and reference architecture. Journal of Reliable Intelligent Environments, 9 (3), 303–317. <https://doi.org/10.1007/s40860-023-00208-6>
- Dirnfeld, R., De Donato, L., Somma, A., Azari, M. S., Marrone, S., Flammini, F., Vittorini, V. (2024). Integrating AI and DTs: challenges and opportunities in railway maintenance application and beyond. Simulation, 100 (9), 903–917. <https://doi.org/10.1177/00375497241229756>
- Bobyl, V., Matusevych, O., Dron, M., Taranenko, A. (2024). The concept of forming a system of change management in the domain of railroad passenger transportation in Ukraine under the conditions of war. Eastern-European Journal of Enterprise Technologies, 1 (13 (127)), 14–21. <https://doi.org/10.15587/1729-4061.2024.297067>
- Thompson, E. A., Lu, P., Alimo, P. K., Atuobi, H. B., Akoto, E. T., Abbew, C. K. (2025). Revolutionizing railway systems: A systematic review of digital twin technologies. High-Speed Railway, 3 (3), 238–250. <https://doi.org/10.1016/j.hspr.2025.05.005>
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Dong, Z., Hou, Y. et al. (2026). A Survey of Large Language Models. Frontiers of Computer Science, 20 (12). <https://doi.org/10.1007/s11704-026-60308-3>
- Lee, K., Joshi, M., Turc, I., Hu, H., Liu, F., Eisenschlos, J. et al. (2023). Pix2Struct: screenshot parsing as pretraining for visual language understanding. Proceedings of the 40th International Conference on Machine Learning (ICML), PMLR 202. Available at: <https://proceedings.mlr.press/v202/lee23g.html>

19. Dong, Y., Jiang, X., Qian, J., Wang, T., Zhang, K., Jin, Z., Li, G. (2025). A survey on code generation with LLM-based agents. arXiv. <https://doi.org/10.48550/arXiv.2508.00083>
20. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J. et al. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18 (6). <https://doi.org/10.1007/s11704-024-40231-1>
21. Bosma, M., Chi, E., Ichter, B., Le, Q. V., Schuurmans, D., Wang, X. et al. (2022). Chain-Of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems* 35, 24824–24837. <https://doi.org/10.52202/068431-1800>
22. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P. et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. <https://doi.org/10.48550/arXiv.2005.14165>
23. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. <https://doi.org/10.48550/arXiv.2005.11401>
24. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y. et al. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55 (12), 1–38. <https://doi.org/10.1145/3571730>
25. Tonmoy, S. M. T. I., Zaman, S. M. M., Jain, V., Rani, A., Rawte, V., Chadha, A., Das, A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models. arXiv. <https://doi.org/10.48550/arXiv.2401.01313>
26. Manakul, P., Liusie, A., Gales, M. (2023). SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9004–9017. <https://doi.org/10.18653/v1/2023.emnlp-main.557>
27. Cancedda, N., Dessi, R., Dwivedi-Yu, J., Hambro, E., Lomeli, M., Raileanu, R. et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems* 36, 68539–68551. <https://doi.org/10.52202/075280-2997>
28. Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q. et al. (2019). Cloud programming simplified: a Berkeley view on serverless computing. Technical Report No. UCB/EECS-2019-3. Berkeley: EECS Department, University of California. Available at: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.html>
29. Shahrad, M., Fonseca, R., Goiri, Í., Chaudhry, G., Batum, P., Cooke, J. et al. (2020). Serverless in the wild: characterizing and optimizing the serverless workload at a large cloud provider. *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, 205–218. Available at: <https://www.usenix.org/conference/atc20/presentation/shahrad>
30. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V. et al. (2017). Serverless Computing: Current Trends and Open Problems. *Research Advances in Cloud Computing*, 1–20. https://doi.org/10.1007/978-981-10-5026-8_1

DOI: 10.15587/1729-4061.2026.357119

DEVELOPMENT OF AN INTERPRETABLE CLASSIFICATION METHOD FOR HEALTHCARE DISINFORMATION DETECTION: LARGE LANGUAGE MODEL FEATURE EXTRACTION AND ENSEMBLE LEARNING (p. 87–100)

Vusal Shahbazov

Institute of Information Technology, Baku, Azerbaijan
ORCID: <https://orcid.org/0009-0009-1758-3082>

Vagif Mammadaliyev

Institute of Information Technology, Baku, Azerbaijan
ORCID: <https://orcid.org/0009-0008-4347-3664>

The object of this study is disinformation in the healthcare domain, distributed through social media and online news sources. Disinformation has become one of the most serious threats to individuals and societies, and it is particularly dangerous when it comes to medical disinformation. The rise of false medical claims online has overwhelmed human researchers, and automated detection methods suffer from several issues, such as low accuracy, the inability to explain solutions derived from implicit deep learning methods or relying on strict text features. This work proposes a method based on large language models (LLMs) and a machine learning (ML) approach for the explainable detection of disinformation in the healthcare sector and identifies attributes inherent in false statements, such as emotionality and rhetorical coloring. Large Language Model Meta AI (LLaMA) serves as a layer for identifying key features, and the ML approach classifies the text with explanations. Shapley Additive Explanations (SHAP) were applied to interpret individual predictions and identify which features contribute most to classification decisions. This method demonstrated high results on two publicly available datasets, achieving an F1 score of approximately 96%. The high performance is explained by the fact that medical disinformation relies on emotional manipulation and persuasive rhetorical patterns, which differ from the neutral tone of reliable medical content. Unlike existing approaches that achieve similar accuracy through opaque methods, the proposed approach relies on interpretable features and provides per-prediction explanations. The proposed method can be applied in automated content-moderation systems and public-health monitoring tools.

Keywords: disinformation detection; medical fake news; sentiment analysis; emotion recognition.

References

1. Wang, Y., McKee, M., Torbica, A., Stuckler, D. (2019). Systematic Literature Review on the Spread of Health-related Misinformation on Social Media. *Social Science & Medicine*, 240, 112552. <https://doi.org/10.1016/j.socscimed.2019.112552>
2. van der Linden, S. (2022). Misinformation: susceptibility, spread, and interventions to immunize the public. *Nature Medicine*, 28 (3), 460–467. <https://doi.org/10.1038/s41591-022-01713-6>
3. Borges do Nascimento, I. J., Beatriz Pizarro, A., Almeida, J., Azzopardi-Muscat, N., André Gonçalves, M. et al. (2022). Infodemics and health misinformation: a systematic review of reviews. *Bulletin of the World Health Organization*, 100 (9), 544–561. <https://doi.org/10.2471/blt.21.287654>
4. Fridman, I., Johnson, S., Elston Lafata, J. (2023). Health Information and Misinformation: A Framework to Guide Research and Practice. *JMIR Medical Education*, 9, e38687. <https://doi.org/10.2196/38687>
5. Suarez-Lledo, V., Alvarez-Galvez, J. (2021). Prevalence of Health Misinformation on Social Media: Systematic Review. *Journal of Medical Internet Research*, 23 (1), e17187. <https://doi.org/10.2196/17187>
6. Cinelli, M., Quattrocioni, W., Galeazzi, A., Valensise, C. M., Brugnoni, E., Schmidt, A. L. et al. (2020). The COVID-19 social media infodemic. *Scientific Reports*, 10 (1). <https://doi.org/10.1038/s41598-020-73510-5>
7. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2017). Fake News Detection on Social Media. *ACM SIGKDD Explorations Newsletter*, 19 (1), 22–36. <https://doi.org/10.1145/3137597.3137600>
8. Choraś, M., Demestichas, K., Gielczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K. et al. (2021). *Advanced Machine Learning*

- techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing*, 101, 107050. <https://doi.org/10.1016/j.asoc.2020.107050>
9. Liu, Z., Zhang, T., Yang, K., Thompson, P., Yu, Z., Ananiadou, S. (2024). Emotion detection for misinformation: A review. *Information Fusion*, 107, 102300. <https://doi.org/10.1016/j.inffus.2024.102300>
 10. Pröllochs, N., Bär, D., Feuerriegel, S. (2021). Emotions explain differences in the diffusion of true vs. false social media rumors. *Scientific Reports*, 11 (1). <https://doi.org/10.1038/s41598-021-01813-2>
 11. Hamed, S. Kh., Ab Aziz, M. J., Yaakub, M. R. (2023). Fake News Detection Model on Social Media by Leveraging Sentiment Analysis of News Content and Emotion Analysis of Users' Comments. *Sensors*, 23 (4), 1748. <https://doi.org/10.3390/s23041748>
 12. Murugesan, S., Pachamuthu, K. (2022). Fake News Detection in the Medical Field Using Machine Learning Techniques. *International Journal of Safety and Security Engineering*, 12 (6), 723–727. <https://doi.org/10.18280/ijssse.120608>
 13. Kolluri, N., Liu, Y., Murthy, D. (2022). COVID-19 Misinformation Detection: Machine-Learned Solutions to the Infodemic. *JMIR Infodemiology*, 2 (2), e38756. <https://doi.org/10.2196/38756>
 14. Iwendi, C., Mohan, S., Khan, S., Ibeke, E., Ahmadian, A., Ciano, T. (2022). Covid-19 fake news sentiment analysis. *Computers and Electrical Engineering*, 101, 107967. <https://doi.org/10.1016/j.compeleceng.2022.107967>
 15. Sikosana, M., Maudsley-Barton, S., Ajao, O. (2025). Advanced Health Misinformation Detection Through Hybrid CNN-LSTM Models Informed by the Elaboration Likelihood Model (ELM). 2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), 1–11. <https://doi.org/10.1109/acdsa65407.2025.11166406>
 16. Alghamdi, J., Lin, Y., Luo, S. (2023). Towards COVID-19 fake news detection using transformer-based models. *Knowledge-Based Systems*, 274, 110642. <https://doi.org/10.1016/j.knosys.2023.110642>
 17. Guo, Y., Ovadje, A., Al-Garadi, M. A., Sarker, A. (2024). Evaluating large language models for health-related text classification tasks with public social media data. *Journal of the American Medical Informatics Association*, 31 (10), 2181–2189. <https://doi.org/10.1093/jamia/ocae210>
 18. Dima, A., Ilis, E., Florea, D., Dascalu, M. (2025). Detection of Fake News in Romanian: LLM-Based Approaches to COVID-19 Misinformation. *Information*, 16 (9), 796. <https://doi.org/10.3390/info16090796>
 19. Bojić, L., Zagovora, O., Zelenkauskaitė, A., Vuković, V., Čabarkapa, M., Veseljević Jerković, S., Jovančević, A. (2025). Comparing large Language models and human annotators in latent content analysis of sentiment, political leaning, emotional intensity and sarcasm. *Scientific Reports*, 15 (1). <https://doi.org/10.1038/s41598-025-96508-3>
 20. Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S. et al. (2021). Fighting an Infodemic: COVID-19 Fake News Dataset. *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, 21–29. https://doi.org/10.1007/978-3-030-73696-5_3
 21. Cui, L., Seo, H., Tabar, M., Ma, F., Wang, S., Lee, D. (2020). DETERRENT: Knowledge Guided Graph Attention Network for Detecting Healthcare Misinformation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 492–502. <https://doi.org/10.1145/3394486.3403092>
 22. Felber, T. (2021). Constraint 2021: Machine Learning Models for COVID-19 Fake News Detection Shared Task. *arXiv*. <https://arxiv.org/abs/2101.03717>
 23. Wani, A., Joshi, I., Khandve, S., Wagh, V., Joshi, R. (2021). Evaluating Deep Learning Approaches for Covid19 Fake News Detection. *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, 153–163. https://doi.org/10.1007/978-3-030-73696-5_15
 24. Karnyoto, A. S., Sun, C., Liu, B., Wang, X. (2022). Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection. *International Journal of Machine Learning and Cybernetics*, 13 (7), 2033–2043. <https://doi.org/10.1007/s13042-021-01503-5>

DOI: 10.15587/1729-4061.2026.363630

EVALUATION OF EDGE-BASED DATA STREAM PROCESSING SCENARIOS IN INDUSTRIAL AUTOMATION SYSTEMS: DATA REDUCTION, LATENCY, AND RESOURCE UTILIZATION (p. 101–110)

Igor Krasnikov

National Technical University «Kharkiv Polytechnic Institute»,
Kharkiv, Ukraine

ORCID: <https://orcid.org/0000-0002-7663-1816>

Kyrylo Halliamov

National Technical University “Kharkiv Polytechnic Institute”,
Kharkiv, Ukraine

ORCID: <https://orcid.org/0009-0007-8762-4885>

Ihor Lysachenko

National Technical University «Kharkiv Polytechnic Institute»,
Kharkiv, Ukraine

ORCID: <https://orcid.org/0000-0002-3723-8587>

This study explores the stream of process measurements after processing at the edge node in the chain “field-level equipment → edge node → SCADA → database”. The task addressed relates to the insufficient quantitative assessment of the impact of processing scenarios at the edge level on the stream intensity, time characteristics, and load of system nodes.

This paper considers an edge-based approach to the stream processing of process data from industrial automation systems. 5 scenarios for processing the stream of process data at the edge node have been considered. A comparative analysis was performed by the number of records after processing, end-to-end latency, reduction ratio, jitter, and resource load. To that end, a bench was implemented that reproduces data transmission from field-level equipment through an edge node to the SCADA system and database. The basic mode without optimization, filtering by the change threshold, periodic selection, aggregation in a time window, event filtering, and a hybrid scheme were investigated.

All edge processing scenarios reduced the stream intensity compared to the baseline. The largest reduction was provided by EV and DB: 0.976 and 0.962. For most scenarios, the end-to-end latency was 71.064–81.197 ms, for HYB – 112.457 ms. The lowest jitter was obtained for EV – 25.98 ms and DB – 38.521 ms. This is explained by the fact that event and threshold selection cut off background signal changes. Preprocessing reduces the network load without a noticeable increase in the load of the edge node. This makes it possible to consider processing at the edge node as a means of data reduction and formation of a controlled stream for the SCADA system and database.

The practical significance implies using the results for comparative selection of processing scenarios for process data taking into account network traffic, latency stability, as well as edge node resources.

Keywords: process data, stream processing, edge processing, SCADA system, stream reduction.

References

1. Wang, S., Wan, J., Zhang, D., Li, D., Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101, 158–168. <https://doi.org/10.1016/j.comnet.2015.12.017>
2. Farooq, M. S., Abdullah, M., Riaz, S., Alvi, A., Rustam, F., Flores, M. A. L. et al. (2023). A Survey on the Role of Industrial IoT in Manufacturing for Implementation of Smart Industry. *Sensors*, 23 (21), 8958. <https://doi.org/10.3390/s23218958>
3. Fragkoulis, M., Carbone, P., Kalavri, V., Katsifodimos, A. (2023). A survey on the evolution of stream processing systems. *The VLDB Journal*, 33 (2), 507–541. <https://doi.org/10.1007/s00778-023-00819-8>
4. Vikash, Mishra, L., Varma, S. (2020). Performance evaluation of real-time stream processing systems for Internet of Things applications. *Future Generation Computer Systems*, 113, 207–217. <https://doi.org/10.1016/j.future.2020.07.012>
5. Kammerer, K., Pryss, R., Hoppenstedt, B., Sommer, K., Reichert, M. (2020). Process-Driven and Flow-Based Processing of Industrial Sensor Data. *Sensors*, 20 (18), 5245. <https://doi.org/10.3390/s20185245>
6. Kumar, R., Agrawal, N. (2023). Analysis of multi-dimensional Industrial IoT (IIoT) data in Edge-Fog-Cloud based architectural frameworks: A survey on current state and research challenges. *Journal of Industrial Information Integration*, 35, 100504. <https://doi.org/10.1016/j.jii.2023.100504>
7. Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., Wu, D. O. (2020). Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges. *IEEE Communications Surveys & Tutorials*, 22 (4), 2462–2488. <https://doi.org/10.1109/comst.2020.3009103>
8. Xhafa, F., Kilic, B., Krause, P. (2020). Evaluation of IoT stream processing at edge computing layer for semantic data enrichment. *Future Generation Computer Systems*, 105, 730–736. <https://doi.org/10.1016/j.future.2019.12.031>
9. Sarasola, T. F. D. B., García, A., Ferrando, J. L. (2024). IIoT Protocols for Edge/Fog and Cloud Computing in Industrial AI. *International Journal of Cloud Applications and Computing*, 14 (1), 1–30. <https://doi.org/10.4018/ijcac.342128>
10. Lu, D., Wang, K., Wang, Y., Shen, Y. (2025). The Proposal and Validation of a Distributed Real-Time Data Management Framework Based on Edge Computing with OPC Unified Architecture and Kafka. *Applied Sciences*, 15 (12), 6862. <https://doi.org/10.3390/app15126862>
11. Krasnikov, I., Hallyamov, K. (2026). Theoretical Foundations of Streaming Processing of Technological Data in Industrial Automation Systems. *Bulletin of the National Technical University “KhPI” A Series of “Information and Modeling”*, 1 (1 (15)), 69–82. <https://doi.org/10.20998/2411-0558.2026.01.04>

DOI: 10.15587/1729-4061.2026.364578

ARCHITECTURAL DESIGN OF GAS TRANSPORTATION AUTOMATION SYSTEM BASED ON THE SYSTEM-TYPE-CENTRIC METHOD (p. 111–138)

Ihor Polataiko

Ivano-Frankivsk National Technical University of Oil and Gas,
Ivano-Frankivsk, Ukraine

ORCID: <https://orcid.org/0000-0002-9111-0162>

Leonid Zamikhovskiy

Ivano-Frankivsk National Technical University of Oil and Gas,
Ivano-Frankivsk, Ukraine

ORCID: <https://orcid.org/0000-0002-6374-8580>

The object of the study is the architectural design process of a gas transportation automation system at the compressor station level as a multi-aspect heterogeneous computer-software system in the automation domain. This system integrates subsystems of different fundamental architectural natures – ranging from PLC programs to cloud-based analytical systems and information systems with operational logic.

The study is focused on solving the problem of completeness, relevance, and interoperability of architectural design artifacts for complex computer-software automation systems. Existing methods predominantly focus on individual aspects – design patterns, system integration and layering, optimal control, or the implementation of specific capabilities, such as supervisory control. They do not form a single, end-to-end architectural design process for the system as a whole.

The essence of the obtained results lies in the formation of an architectural model of the system through the execution of the architectural design process using a system-type-centric method. The constructed architectural model is distinguished by the traceability of architectural artifacts and the selection of architectural viewpoints depending on the fundamental nature of the systems, which ensures the coverage of fundamental architectural aspects and achieves the completeness of the architectural description. It is built through the sequential formation of system views from relevant viewpoints.

The results demonstrate the scientific and practical feasibility of the method for the comprehensive design of complex heterogeneous automation systems that combine operational technology (OT) and information technology (IT) levels. From a practical standpoint, the obtained results can be applied as a reference architectural model for the implementation of the method.

Keywords: architectural design, automation, gas transportation, system-type-centric paradigm.

References

1. Williams, T. J. (1990). A Reference Model for Computer Integrated Manufacturing from the Viewpoint of Industrial Automation. *IF-AC Proceedings Volumes*, 23 (8), 281–291. [https://doi.org/10.1016/s1474-6670\(17\)51748-6](https://doi.org/10.1016/s1474-6670(17)51748-6)
2. Polataiko, I., Zamikhovskiy, L. (2026). Development of system-type-centric paradigm of computer-software systems architectural design for automation systems. *Technology Audit and Production Reserves*, 1 (2 (87)), 43–56. <https://doi.org/10.15587/2706-5448.2026.349943>
3. Polataiko, I., Zamikhovskiy, L. (2026). Development of system-type-centric method for architectural design of computer-software automation systems. *Eastern-European Journal of Enterprise Technologies*, 2 (2 (140)), 65–84. <https://doi.org/10.15587/1729-4061.2026.359147>
4. Gorbijchuk, M. I., Lazoriv, O. T., Lazoriv, A. M. (2020). The computer system for optimal control of natural gas pumping units. *Methods and Devices of Quality Control*, 2 (45), 90–101. [https://doi.org/10.31471/1993-9981-2020-2\(45\)-90-101](https://doi.org/10.31471/1993-9981-2020-2(45)-90-101)
5. Damirova, J., Salmanov, V. (2026). Control of compressor and pumping systems in the oil and gas industry using SCADA. *ETM Equipment Technologies Materials*, 217. <https://doi.org/10.36962/etm33022026-25>
6. Zamikhovskiy, L., Nykolaychuk, M., Levytskyi, I. (2025). Extending the functionality of topologies of Web-oriented control systems for technological objects based on “Open User Communication”. *Eastern-European Journal of Enterprise Technologies*, 6 (2 (138)), 94–115. <https://doi.org/10.15587/1729-4061.2025.348728>

7. Zamikhovskiy, L., Zamikhovska, O., Ivanyuk, N., Mirzoieva, O., Nykolaychuk, M. (2025). Development of an anti-surge protection system for gas pumping units based on hardware and software vibration monitoring tools. *Eastern-European Journal of Enterprise Technologies*, 4 (2 (136)), 117–132. <https://doi.org/10.15587/1729-4061.2025.337736>
8. Zamikhovskiy, L., Nykolaychuk, M., Levytskyi, I. (2025). Development of a simulation model of a WEB-oriented servo drive frequency control system based on “Digital Twins” technology. *Technology Audit and Production Reserves*, 6 (2 (86)), 76–90. <https://doi.org/10.15587/2706-5448.2025.345825>
9. Nykolaychuk, M., Zamikhovskiy, L., Levytskyi, I., Kopei, V., Ropyak, L. (2026). Development of a Simulation Model of a PID Controller Based on Simatic S7 Hardware-Software Tools and “Digital Twin” Technology. *Automation*, 7 (3), 74. <https://doi.org/10.3390/automation7030074>
10. IEC 62264-1:2013. Enterprise-control system integration - Part 1: Models and terminology. International Electrotechnical Commission. Available at: <https://webstore.iec.ch/en/publication/6675>
11. The Industrial Internet Reference Architecture (IIRA), Version 1.10: An Industry IoT Consortium foundational document (2022). Boston: Industry IoT Consortium. Available at: <https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf>
12. Megow, J. (2020). Reference architecture models for Industry 4.0, smart manufacturing and IoT: An introduction. Berlin: Begleitforschung PAiCE; IIT – Institut für Innovation und Technik in der VDI / VDE Innovation + Technik GmbH. Available at: https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/PAiCE_Leitfaden_Reference_Architecture.pdf?__blob=publicationFile&v=1
13. Villafuerte, C., Moncayo, M., Oñate, W. (2026). RAMI 4.0 Architecture for Industrial Traceability with Artificial Intelligence and Integrated Security. *Automation*, 7 (3), 72. <https://doi.org/10.3390/automation7030072>
14. Folgado, F., Calderón, D., González, I., Calderón, A. (2024). Review of Industry 4.0 from the Perspective of Automation and Supervision Systems: Definitions, Architectures and Recent Trends. *Electronics*, 13 (4), 782. <https://doi.org/10.3390/electronics13040782>
15. Starke, G., Simons, M., Zörner, S., Müller, R. D., Losch, H. (2019). *arc42 by Example: Software architecture documentation in practice*. Birmingham: Packt Publishing.
16. Brown, S. (2015). *The C4 model for visualising software architecture*. Leanpub. Available at: <https://leanpub.com/visualising-software-architecture>
17. Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software*, 12 (6), 42–50. <https://doi.org/10.1109/52.469759>
18. Bass, L., Clements, P., Kazman, R. (2021). *Software Architecture in Practice*. Addison-Wesley Professional.
19. Cervantes, H., Kazman, R. (2016). *Designing Software Architectures: A Practical Approach*. Boston: Addison-Wesley Professional. Available at: https://repo.darmajaya.ac.id/3948/1/Designing%20Software%20Architectures_%20A%20Practical%20Approach%20%28%20PDFDrive%20%29.pdf
20. Rozanski, N., Woods, E. (2011). *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Boston: Addison-Wesley. Available at: <https://ptgmedia.pearsoncmg.com/images/9780321718334/samplepages/032171833X.pdf>
21. Calderón, D., Folgado, F. J., González, I., Calderón, A. J. (2024). Implementation and Experimental Application of Industrial IoT Architecture Using Automation and IoT Hardware/Software. *Sensors*, 24 (24), 8074. <https://doi.org/10.3390/s24248074>
22. Villar, E., Martín Toral, I., Calvo, I., Barambones, O., Fernández-Bustamante, P. (2024). Architectures for Industrial AIoT Applications. *Sensors*, 24 (15), 4929. <https://doi.org/10.3390/s24154929>
23. Gouriseti, S. N. G., Bhadra, S., Sebastian-Cardenas, D. J., Touhiduzzaman, M., Ahmed, O. (2023). A Theoretical Open Architecture Framework and Technology Stack for Digital Twins in Energy Sector Applications. *Energies*, 16 (13), 4853. <https://doi.org/10.3390/en16134853>
24. Lamm, J. G., Weikens, T. (2013). Method for Deriving Functional Architectures from Use Cases. *Systems Engineering*, 17 (2), 225–236. <https://doi.org/10.1002/sys.21265>
25. System Architecture Framework. (GfSE). Available at: <https://saf.gfse.org/>
26. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison-Wesley Professional. Available at: <https://fabiofumarola.github.io/nosql/readingMaterial/Evans03.pdf>
27. Waqas, M., Jamil, M., Khan, A. A. (2024). Hybrid Power System Design and Dynamic Modeling for Enhanced Reliability in Remote Natural Gas Pipeline Control Stations. *Energies*, 17 (7), 1763. <https://doi.org/10.3390/en17071763>
28. Nazarenko, I. V., Nikolaychuk, M. Ya., Ferenets, V. D., Sukhanov, D. Ye. (2014). Construction and modeling of unified control systems of actuating mechanisms for objects of gas-transport system. *Eastern-European Journal of Enterprise Technologies*, 1 (2 (67)), 41–48. <https://doi.org/10.15587/1729-4061.2014.21204>
29. Waqas, M., Jamil, M. (2024). Smart IoT SCADA System for Hybrid Power Monitoring in Remote Natural Gas Pipeline Control Stations. *Electronics*, 13 (16), 3235. <https://doi.org/10.3390/electronics13163235>
30. Sayghe, A. (2025). Digital Twin-Driven Intrusion Detection for Industrial SCADA: A Cyber-Physical Case Study. *Sensors*, 25 (16), 4963. <https://doi.org/10.3390/s25164963>
31. Zamikhovskiy, L., Nykolaychuk, M., Levytskyi, I. (2026). Method for building a virtual simulator of the drowworks automated control subsystem of a drilling rig with a WEB interface. *IOP Conference Series: Earth and Environmental Science*, 1630 (1), 12068. <https://doi.org/10.1088/1755-1315/1630/1/012068>
32. Chen, S., Ebe, F., Morris, J., Lorenz, H., Kondzialka, C., Heilscher, G. (2022). Implementation and Test of an IEC 61850-Based Automation Framework for the Automated Data Model Integration of DES (ADMID) into DSO SCADA. *Energies*, 15 (4), 1552. <https://doi.org/10.3390/en15041552>
33. Cirani, S., Ferrari, G., Mancin, M., Picone, M. (2018). Virtual Replication of IoT Hubs in the Cloud: A Flexible Approach to Smart Object Management. *Journal of Sensor and Actuator Networks*, 7 (2), 16. <https://doi.org/10.3390/jsan7020016>
34. Rosioru, S., Mihai, V., Neghina, M., Craciunean, D., Stamatescu, G. (2022). PROSIM in the Cloud: Remote Automation Training Platform with Virtualized Infrastructure. *Applied Sciences*, 12 (6), 3038. <https://doi.org/10.3390/app12063038>
35. Dobaj, J., Riel, A., Macher, G., Egretzberger, M. (2023). Towards DevOps for Cyber-Physical Systems (CPSs): Resilient Self-Adaptive Software for Sustainable Human-Centric Smart CPS Facilitated by Digital Twins. *Machines*, 11 (10), 973. <https://doi.org/10.3390/machines11100973>
36. Solomchak, O., Nykolaychuk, M., Solomchak, A. (2024). Modeling and Simulation of Distribution Network with the D-STATCOM and Photovoltaic Power Stations Using MATLAB. 2024 IEEE 5th KhPI Week on Advanced Technology (KhPIWeek), 1–5. <https://doi.org/10.1109/khpiweek61434.2024.10878105>

37. Petrescu, I., Niculae, E., Vulturescu, V., Dimitrescu, A., Ungureanu, L. M. (2025). Transport and Application Layer Protocols for IoT: Comprehensive Review. *Technologies*, 13 (12), 583. <https://doi.org/10.3390/technologies13120583>
38. Christ, L., Milloch, E., Boshoff, M., Hypki, A., Kuhlenkötter, B. (2023). Implementation of Digital Twin and Real Production System to Address Actual and Future Challenges in Assembly Technology. *Automation*, 4 (4), 345–358. <https://doi.org/10.3390/automation4040020>
39. Zamikhovskiy L., Nykolaychuk, M., Levitskiy, I. (2026). Development of a simulation model and testing methodology for frequency control systems of an induction drives using Sinamics G220 and Digital Twin technology. *Technology Audit and Production Reserves*, 3 (2 (89)), 75–90. <https://doi.org/10.15587/2706-5448.2026.360983>
40. Al-Najari, B., Hen, C. K., Siaw Paw, J. K., Marhoon, A. F. (2025). Dynamic Tuning of PLC-Based Built-In PID Controller Using PSO-MANFIS Hybrid Algorithm via OPC Server. *Automation*, 6 (4), 83. <https://doi.org/10.3390/automation6040083>
41. Rahman, M. A., Shahrior, M. F., Iqbal, K., Abushaiba, A. A. (2025). Enabling Intelligent Industrial Automation: A Review of Machine Learning Applications with Digital Twin and Edge AI Integration. *Automation*, 6 (3), 37. <https://doi.org/10.3390/automation6030037>

DOI: 10.15587/1729-4061.2026.359626

DESIGNING AN ADAPTIVE-PREDICTIVE SYSTEM TO CONTROL HYDRODYNAMIC MODES OF OIL TRANSPORTATION (p. 139–152)

Oleksandr Kuchmystenko

Ivano-Frankivsk National Technical University of Oil and Gas,
Ivano-Frankivsk, Ukraine

ORCID: <https://orcid.org/0000-0002-0457-7611>

Mykhailo Shavranskyi

Ivano-Frankivsk National Technical University of Oil and Gas,
Ivano-Frankivsk, Ukraine

ORCID: <https://orcid.org/0009-0007-9858-9202>

Yurii Puk

Ivano-Frankivsk National Technical University of Oil and Gas,
Ivano-Frankivsk, Ukraine

ORCID: <https://orcid.org/0009-0008-5297-1358>

This study investigates the process of automated control over the operation of pumping stations along the main oil pipeline during start-up, stop, and pressure maintenance while oil pumping. Disturbances are considered as deterministic parametric variations of the coefficient of hydraulic resistance, which depend on changes in pressures, flows, including oil viscosity.

A control system has been designed that solves the problem of integrating recursive identification of model parameters with predictive formation of control effects in a single loop that covers start-up, operating mode, and compensation for disturbances. The main idea of automation is to transfer the functions of controlling the oil pipeline from pumping station operators to the dispatch center.

An adaptive-predictive control structure has been proposed that combines recursive identification of parameters, state assessment, and optimization predictive formation of control effects. The system integrates a Kalman filter for real-time state assessment and a module for recursive updating of model parameters. This provides online adaptation of the mathematical model of the facility without personnel intervention.

The application of the proposed approach reduces the accumulation of the integrated error in flow control under the operating mode by approximately 87 percent. After a viscosity jump, the flow does not go beyond the technological tolerance of ± 2 percent. The improvement in control quality is explained by the system's capability to predict the dynamics of the facility and compensate for uncertainties before their impact on the controlled variables. The algorithm automatically processes the full cycle: controlled start-up, entry into the operating mode, and maintaining pressures within the specified limits.

The results could be implemented in SCADA systems for oil pumping stations to be controlled from the dispatch center. This would increase the economic efficiency and stability of oil transportation modes.

Keywords: Kalman filter, recursive identification, control over hydrodynamic modes, adaptive control, predictive control, mathematical model.

References

1. Chaudhry, M. H. (2014). *Applied Hydraulic Transients*. Springer New York. <https://doi.org/10.1007/978-1-4614-8538-4>
2. Zeghadnia, L., Robert, J. L., Achour, B. (2019). Explicit solutions for turbulent flow friction factor: A review, assessment and approaches classification. *Ain Shams Engineering Journal*, 10 (1), 243–252. <https://doi.org/10.1016/j.asej.2018.10.007>
3. Tentis, E., Margaris, D., Papanikas, D. (2003). Transient gas flow simulation using an Adaptive Method of Lines. *Comptes Rendus. Mécanique*, 331 (7), 481–487. [https://doi.org/10.1016/s1631-0721\(03\)00106-2](https://doi.org/10.1016/s1631-0721(03)00106-2)
4. Muñoz, J. A. D., Ancheyta, J., Castañeda, L. C. (2016). Required Viscosity Values To Ensure Proper Transportation of Crude Oil by Pipeline. *Energy & Fuels*, 30 (11), 8850–8854. <https://doi.org/10.1021/acs.energyfuels.6b01908>
5. Azizi, N., Homayoon, R., Hojjati, M. R. (2018). Predicting the Colebrook–White Friction Factor in the Pipe Flow by New Explicit Correlations. *Journal of Fluids Engineering*, 141 (5). <https://doi.org/10.1115/1.4041232>
6. Simon, D. (2006). *Optimal State Estimation*. Wiley. <https://doi.org/10.1002/0470045345>
7. Simon, D. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4 (8), 1303–1318. <https://doi.org/10.1049/iet-cta.2009.0032>
8. Borase, R. P., Maghade, D. K., Sondkar, S. Y., Pawar, S. N. (2020). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, 9 (2), 818–827. <https://doi.org/10.1007/s40435-020-00665-4>
9. Wang, L. (2004). A Tutorial on Model Predictive Control: Using a Linear Velocity-Form Model. *Developments in Chemical Engineering and Mineral Processing*, 12 (5-6), 573–614. <https://doi.org/10.1002/apj.5500120511>
10. Qin, S. J., Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11 (7), 733–764. [https://doi.org/10.1016/s0967-0661\(02\)00186-7](https://doi.org/10.1016/s0967-0661(02)00186-7)
11. Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36 (6), 789–814. [https://doi.org/10.1016/s0005-1098\(99\)00214-9](https://doi.org/10.1016/s0005-1098(99)00214-9)
12. Zeng, W., Wang, C., Yang, J. (2022). Hydraulic Transient Simulation of Pipeline-Open Channel Coupling Systems and Its Applications in

- Hydropower Stations. *Water*, 14 (18), 2897. <https://doi.org/10.3390/w14182897>
13. Allgöwer, F., Zheng, A. (Eds.) (2000). *Nonlinear Model Predictive Control*. Birkhäuser Basel. <https://doi.org/10.1007/978-3-0348-8407-5>
 14. Gopalakrishnan, A., Kaisare, N. S., Narasimhan, S. (2011). Incorporating delayed and infrequent measurements in Extended Kalman Filter based nonlinear state estimation. *Journal of Process Control*, 21 (1), 119–129. <https://doi.org/10.1016/j.jprocont.2010.10.013>
 15. Ljung, L. (1998). *System Identification. Signal Analysis and Prediction*, 163–173. https://doi.org/10.1007/978-1-4612-1768-8_11
 16. Ioannou, P., Fidan, B. (2006). *Adaptive Control Tutorial*. SIAM. <https://doi.org/10.1137/1.9780898718652>
 17. Gülich, J. F. (2014). *Centrifugal Pumps*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-40114-5>
 18. Bikmukhametov, T., Jäschke, J. (2020). Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers & Chemical Engineering*, 138, 106834. <https://doi.org/10.1016/j.compchemeng.2020.106834>
 19. Nelles, O. (2001). *Nonlinear System Identification*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-04323-3>
 20. Camacho, E. F., Bordons, C. (2007). *Model Predictive control*. Springer London. <https://doi.org/10.1007/978-0-85729-398-5>
 21. Pannocchia, G., Rawlings, J. B. (2003). Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49 (2), 426–437. <https://doi.org/10.1002/aic.690490213>

DOI: 10.15587/1729-4061.2026.357361

УДОСКОНАЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СИНТЕЗУ ПАРАЛЕЛЬНО-ПОТОКОВИХ СТРУКТУР ВЕРТИКАЛЬНО-ГРУПОВОГО ОБЧИСЛЕННЯ БАГАТООПЕРАНДНИХ НЕЙРООПЕРАЦІЙ У РЕАЛЬНОМУ ЧАСІ (с. 6–23)

І. Г. Цмоць, В. М. Теслюк, Ю. В. Опотяк, Б. В. Штогрінець

У дослідженні розглянуто підхід до удосконалення інформаційної технології синтезу структур для опрацювання даних багатопарандних нейрооперацій у реальному часі. Проблема полягає у відсутності формалізованого підходу до синтезу таких структур здатного одночасно враховувати параметри потоків даних, глибину конвеєра, ступінь паралелізму та апаратні обмеження при забезпеченні заданих часових характеристик. Розроблено методи паралельного вертикально-групового обчислення скалярного добутку, суми квадратів різниць та пошуку максимального і мінімального значень. Удосконалено метод конкретизації поточкових графів. Розроблено базові паралельно-поточкові обчислювальні структури та аналітичні вирази для оцінювання апаратних витрат, тривалості конвеєрного такту і ефективності використання апаратних ресурсів. На цій основі у роботі удосконалено інформаційну технологію синтезу паралельно-поточкових структур вертикально-групового обчислення багатопарандних нейрооперацій у реальному часі. Поставлена проблема вирішена шляхом поєднання вертикальної і групової паралельності, конвеєризації, модульної організації, узгодженню інтенсивності надходження даних з інтенсивністю їх опрацювання та поетапному переходу від алгоритмічного опису до апаратної реалізації. Удосконалена інформаційна технологія забезпечує зменшення апаратних витрат, підвищення пропускної здатності, зниження латентності та вибір оптимальних параметрів структур. Одночасне опрацювання груп розрядних зрізів зменшує кількість конвеєрних сходинок, а конкретизація поточкових графів дає змогу адаптувати структуру обчислень до вимог реального часу. Практично результати можуть бути використані під час синтезу спеціалізованих FPGA-, ASIC-, SoC-засобів для нейроорієнтованих систем реального часу із заданими характеристиками.

Ключові слова: поточкові графи, системи реального часу, апаратні витрати, ефективність використання ресурсів.

DOI: 10.15587/1729-4061.2026.360407

РОЗРОБКА МЕТОДИКИ ПОРІВНЯННЯ МЕТОДІВ ВИБОРУ ОПТИМАЛЬНОЇ ВАРІАЦІЇ АРХІТЕКТУРИ РОЗМЕЖУВАННЯ ВІДПОВІДАЛЬНОСТІ КОМАНД І ЗАПИТІВ У ПОЄДНАННІ З ПІДХОДОМ ФІКСАЦІЇ ТА ЗБЕРЕЖЕННЯ ПОДІЙ (с. 24–44)

Д. Л. Грузін, О. А. Литвинов

Об'єктом дослідження є порівняння методів оцінювання архітектури програмного забезпечення. Дослідження спрямовано на підвищення об'єктивності такого порівняння за рахунок використання кількісних показників, отриманих на основі параметричної моделі, адекватність якої перевірено експериментально.

У роботі запропоновано методику кількісного порівняння методів оцінювання архітектури програмного забезпечення. За допомогою цієї методики здійснено порівняння методу аналізу архітектурних компромісів (англ. Architectural Trade-off Analysis Method, ATAM) та технології підтримки прийняття рішень щодо архітектурних варіацій архітектури розмежування відповідальності команд і запитів (Command Query Responsibility Segregation, CQRS) з підходом фіксації та збереження подій (Event Sourcing, ES) (англ. Decision-making Support regarding CQRS with ES Architectural Variations, DSAV-CQRSES).

Особливостями методики є використання параметричної моделі, побудованої на основі аналізу реальних проєктів, що дає змогу обґрунтувати коректність референтної варіації, та кількісне оцінювання методів, зокрема за тривалістю застосування та розбіжністю результатів.

Запропонована методика дала змогу кількісно підтвердити переваги DSAV-CQRSES над ATAM для розв'язання задачі вибору оптимальної варіації архітектури CQRS з ES.

Тривалість застосування ATAM становила 32 години за участю чотирьох фахівців (загальні витрати 88 людино-годин). Технологія DSAV-CQRSES застосовувалася одним фахівцем упродовж 40 годин. За середніх і граничних параметрів моделі DSAV-CQRSES стабільно визначала рекомендовану варіацію з коефіцієнтом варіації 2,8%. Натомість ATAM продемонстрував нижчу точність і забезпечив коректний вибір у трьох із п'яти випадків за коефіцієнта варіації 33,64%.

Результати експерименту можуть бути використані на практиці для вибору інструментів порівняння варіацій архітектури CQRS з ES.

Ключові слова: оптимізація, обчислювальний експеримент, ATAM, DSAV-CQRSES, аналіз даних, порівняльний аналіз.

DOI: 10.15587/1729-4061.2026.362735

РОЗРОБКА МЕТОДУ КРОС-ПРОЄКТНОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НЕЧІТКИХ АЛГОРИТМІЧНИХ МОДЕЛЕЙ (с. 45–55)

Г. Б. Ракитянська, Б. В. Прус

Об'єктом дослідження є нечіткі алгоритмічні моделі прогнозування дефектів програмного забезпечення. Проблемою традиційних моделей машинного (глибокого) навчання залишаються складність та інтерпретабельність. Крім того, крос-проєктне навчання таких моделей потребує вирішення проблеми неоднорідності вихідного та цільового розподілів даних. Запропоновано нечітку

алгоритмічну модель прогнозування дефектів програмного забезпечення з інтегрованим розподілом варіантів реалізації етапів розробки. Розроблено генетично-нейронний метод налаштування нечіткої алгоритмічної моделі на крос-проектні дані. На відміну від моделей машинного навчання, інтерпретабельність моделі прогнозування досягнуто за рахунок оцінки правильності (дефектності) етапів розробки за допомогою нечітких правил. Модель будується на основі алгоритмічних структур «робота-контроль-доробка», що є аналогом нечіткої бази знань. Налаштуванню підлягають нечіткі оцінки дефектності виконання робочих, контрольних та доробочних операторів, де ранги дефектів моделюють розподіл ресурсів. Навчальна вибірка складається на основі оцінок правильності варіантів реалізації алгоритмічних структур. Вбудова індикаторів варіантів реалізації робочих, контрольних та доробочних операторів дозволила вирішити проблему неоднорідності вихідного та цільового розподілів даних. На відміну від відомих методів навчання нейро-нечітких моделей надійності програмного забезпечення, спрощення процесу налаштування досягнуто за рахунок трансферу крос-проектних даних. Сферою практичного застосування є прогнозування якості нового програмного продукту на основі досвіду завершених проектів. Умовою використання є дискретна алгоритмічна модель процесу розробки.

Ключові слова: прогнозування дефектів програмного забезпечення, нечітка алгоритмічна модель, крос-проектне трансферне навчання.

DOI: 10.15587/1729-4061.2026.360693

РОЗРОБЛЕННЯ МЕТОДУ АВТОМАТИЗОВАНОЇ ІНТЕГРАЦІЇ МОДЕЛІ ТЕСТУВАННЯ «ПЕНТАГОН» В КОНВЕЄР БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ТА ДОСТАВКИ ДЛЯ МІКРОСЕРВІСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (с. 56–66)

О. М. Кузьмич, М. М. Сенів

Об'єктом дослідження є процес верифікації надійності програмного забезпечення з мікросервісною архітектурою, яке розробляється паралельно декількома командами. Проблема полягає у відсутності адаптованих до мікросервісної архітектури, економічно доцільних, методів автоматизованої інтеграції моделей тестування у конвеєр безперервної інтеграції та доставки (CI/CD). Це призводить до виникнення «сліпих зон» у сфері функціональних та нефункціональних вимог та надмірні витрати часу розробників. У роботі розроблено метод автоматизованої інтеграції моделі тестування «Пентагон» у багатоступеневий конвеєр безперервної інтеграції та доставки. Метод реалізує кумулятивний принцип прогресивного розподілу шести шарів тестування між чотирма ізольованими середовищами: DEV (розробки), TEST (тестове), PRE-PROD (перед продуктове) та PROD (продуктове). Розроблений метод гарантує покриття всіх шести шарів під час кожної зміни в кодовій базі. Ключова особливість методу полягає у виключенні впливу людського фактору на прийняття рішень щодо запуску тестів. Це дозволяє досягти підвищення значень надійності, а саме Defect Removal Efficiency (DRE), тоді як ручний метод виявляє в середньому лише 27% дефектів. Розроблений метод скорочує час, коли розробник задіяний в процесі запуску тестів на 70% порівняно з ручним підходом. За 100 комітів (фіксація поточного стану вашого коду та файлів у локальному репозиторії) на місяць економія порівняно з ручним методом становить 37%. Метод придатний для верифікації програмного забезпечення з мікросервісною архітектурою, яке розгортається на хмарних платформах (Amazon Web Services, США; Azure, США; Google Cloud, США.) з підтримкою автоматизованого тестування та контейнерної оркестрації.

Ключові слова: мікросервісна архітектура, конвеєр безперервної інтеграції та доставки, надійність програмного забезпечення.

DOI: 10.15587/1729-4061.2026.363077

РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБЗАСТОСУНКІВ НА JAVA ТА SPRING МЕТОДОМ РЕГРЕСІЙНОГО АНАЛІЗУ (с. 67–74)

Л. М. Макарова, Л. О. Латанська, А. В. Пухалевич, В. О. Каіров, М. О. Джурицький

Об'єктом дослідження є процес оцінювання розміру вебзастосунків, розроблених мовою Java (країна походження – США) з використанням фреймворку Spring (країна походження – США).

Проблемою, що вирішувалась у роботі, є підвищення достовірності оцінювання розміру відповідних вебзастосунків. Оцінювання розміру вебзастосунків, розроблених мовою Java з використанням фреймворку Spring, є важливою задачею інженерії програмного забезпечення. Це дозволяє оцінювати необхідні часові рамки для реалізації функціоналу і планувати бюджет проекту.

В результаті роботи була побудована математична модель для оцінювання розміру вебзастосунків на Java та Spring на основі нормалізованого перетворення десяткового логарифма, та розроблена відповідна програма для автоматизації розрахунків. Параметри якості побудованої моделі: $R^2 = 0.9173$, $MMRE = 0.1511$, $PRED(0.25) = 0.7931$.

Це дозволило підвищити достовірність оцінювання розміру таких вебзастосунків, а саме: зменшити значення середньої величини відносної похибки та підвищити рівень прогнозування, а також звужити ширину довірчого та прогнозного інтервалів.

Був зібраний набір даних, що складається із 36 проектів, та виконана його передобробка, яка включала перевірку на багатовимірну нормальність розподілу та нормалізацію шляхом логарифмування. Була побудована відповідна нелінійна регресійна модель. Для підвищення достовірності моделі був застосований алгоритм, що включає ітераційне вилучення викидів за допомогою квадрата відстані Махаланобіса та критерію Фішера. Розроблено програму на основі побудованої нелінійної регресійної моделі та проведено аналіз отриманих результатів.

Аналіз якості побудованої моделі свідчить про її адекватність та можливість застосування для вирішення задач оцінювання розміру відповідного програмного забезпечення.

Ключові слова: оцінювання розміру, нормалізація, нелінійна регресія, викид, Java, Spring, програмне забезпечення.

DOI: 10.15587/1729-4061.2026.363741

РОЗРОБЛЕННЯ ДВОРІВНЕВОЇ АГЕНТНОЇ АРХІТЕКТУРИ ДЛЯ АВТОМАТИЗОВАНОГО ГЕНЕРУВАННЯ ТА ВАЛІДАЦІЇ СХЕМ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ВАГОНІВ З ВИКОРИСТАННЯМ МУЛЬТИМОДАЛЬНИХ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ (с. 75–86)

Ivan Dobrovolskyi

Об'єктом дослідження є автоматизоване генерування та валідація схем інтерфейсу вагонів у системах бронювання залізничних квитків. Ручне оновлення однієї конфігурації займає близько 24 годин, залежить від релізів, потребує участі інженерів і підвищує ризик виробничих помилок.

Запропоновано дворівневу агентну архітектуру. Вона включає розмежування міркування й виконання, мультимодальне перетворення схем у структуровані конфігурації та детерміністичну валідацію. Архітектуру перевірено на платформі Укрзалізниці, що підтримує понад 150 конфігурацій вагонів і обслуговує понад 20 млн пасажирів на рік.

Опрацьовано 20 виробничих схем різних класів вагонів. Одна схема потребувала ручного виправлення. Інші були уточнені одним або двома додатковими запитами. Час створення нового типу вагона зменшився з приблизно 24 годин до 15 хвилин. Скорочення становить близько 99%. Щомісячні інженерні витрати знизилися з тисяч доларів до менш ніж 5 доларів. Інцидентів, спричинених галуцинаціями моделі, не зафіксовано.

Рішення дає змогу працювати з понад 150 конфігураціями без завантаження всіх схем в один контекст. Воно дозволяє нетехнічним адміністраторам оновлювати конфігурації поза релізними циклами, зменшує залежність від інженерів і може застосовуватися для структурованого цифрового представлення фізичних об'єктів. Результати показують можливість застосування агентного AI у критичних цифрових сервісах за умови обов'язкової валідації перед впровадженням. Запропонований підхід не замінює перевірку, а включає її до процесу створення конфігурацій. Це скорочує ручну інженерну роботу без зміни основного програмного коду. Частина операційних дій може виконуватися нетехнічними адміністраторами, тоді як контроль структури, формату й коректності залишається обов'язковим. Підхід може бути використаний у залізничному квиткуванні та суміжних системах, де фізичні об'єкти подаються як цифрові інтерфейсні схеми.

Ключові слова: автоматизація квиткування, розмежування міркування й виконання, виробнича валідація, комп'ютерний зір, безсерверне розгортання.

DOI: 10.15587/1729-4061.2026.357119

РОЗРОБКА ІНТЕРПРЕТОВАНОГО МЕТОДУ КЛАСИФІКАЦІЇ ДЛЯ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ В ОХОРОНІ ЗДОРОВ'Я: ВИЛУЧЕННЯ ОЗНАК З ВЕЛИКОЇ МОВНОЇ МОДЕЛІ ТА АНСАМБЛЕВЕ НАВЧАННЯ (с. 87–100)

Vusal Shahbazov, Vagif Mammadaliyev

Об'єктом цього дослідження є дезінформація в сфері охорони здоров'я, що поширюється через соціальні мережі та онлайн-джерела новин. Дезінформація стала однією з найсерйозніших загроз для окремих осіб та суспільств, і вона особливо небезпечна, коли йдеться про медичну дезінформацію. Зростання кількості неправдивих медичних заяв в Інтернеті приголомшило дослідників, а автоматизовані методи виявлення страждають від кількох проблем, таких як низька точність, нездатність пояснити рішення, отримані за допомогою методів неявного глибокого навчання або спираючись на суворі текстові характеристики. У цій роботі пропонується метод, заснований на великих мовних моделях (LLM) та підході машинного навчання (ML) для поясненого виявлення дезінформації в секторі охорони здоров'я, та визначає атрибути, властиві хибним твердженням, такі як емоційність та риторичне забарвлення. Мета-штучний інтелект великої мовної моделі (LLaMA) служить шаром для визначення ключових характеристик, а підхід ML класифікує текст за допомогою пояснень. Для інтерпретації окремих прогнозів та визначення того, які характеристики найбільше сприяють прийняттю рішень щодо класифікації, були застосовані адитивні пояснення Шеплі (SHAP). Цей метод продемонстрував високі результати на двох загальнодоступних наборах даних, досягнувши балу F1 приблизно 96%. Висока ефективність пояснюється тим, що медична дезінформація спирається на емоційні маніпуляції та переконливі риторичні моделі, які відрізняються від нейтрального тону надійного медичного контенту. На відміну від існуючих підходів, які досягають подібної точності за допомогою непрозорих методів, запропонований підхід спирається на інтерпретовані ознаки та надає пояснення на основі прогнозів. Запропонований метод може бути застосований в автоматизованих системах модерації контенту та інструментах моніторингу громадського здоров'я.

Ключові слова: виявлення дезінформації; медичні фейкові новини; аналіз настроїв; розпізнавання емоцій.

DOI: 10.15587/1729-4061.2026.363630

ВИЗНАЧЕННЯ ВПЛИВУ СЦЕНАРІЇВ ПЕРИФЕРІЙНОЇ ПОТОКОВОЇ ОБРОБКИ ДАНИХ У ПРОМИСЛОВИХ СИСТЕМАХ АВТОМАТИЗАЦІЇ: РЕДУКЦІЯ ДАНИХ, ЗАТРИМКА ТА ВИКОРИСТАННЯ РЕСУРСІВ (с. 101–110)

I. Л. Красніков, К. С. Галлямов, І. Г. Лисаченко

Об'єктом дослідження є потік технологічних вимірювань після обробки на периферійному вузлі в ланцюгу «обладнання польового рівня → периферійний вузол → SCADA → база даних». Проблема полягає в недостатній кількісній оцінці впливу сценаріїв обробки на периферійному рівні на інтенсивність потоку, часові характеристики та навантаження вузлів системи. У роботі розглядається периферійний підхід до потокової обробки технологічних даних промислових систем автоматизації. Розглянуто 5 сценаріїв обробки потоку технологічних даних на периферійному вузлі. Порівняльний аналіз виконано за кількістю записів після обробки, наскрізною затримкою, коефіцієнтом редукції, варіативністю затримки та ресурсним навантаженням. Для цього реалізовано стенд, що відтворює

передавання даних від обладнання польового рівня через периферійний вузол до SCADA-системи і бази даних. Досліджено базовий режим без оптимізації, фільтрацію за порогом зміни, періодичний відбір, агрегацію у часовому вікні, подієву фільтрацію та гібридну схему. Усі сценарії периферійної обробки зменшили інтенсивність потоку порівняно з базовим режимом. Найбільшу редукацію забезпечили EV і DB: 0.976 та 0.962. Для більшості сценаріїв наскрізна затримка становила 71.064–81.197 мс, для НУВ – 112.457 мс. Найменшу варіативність затримки отримано для EV – 25.98 мс і DB – 38.521 мс. Це пояснюється тим, що подієва та порогова селекція відсікають фонові зміни сигналу. Попередня обробка знижує мережеве навантаження без помітного збільшення навантаження периферійного вузла. Це дозволяє розглядати обробку на периферійному вузлі як засіб скорочення даних і формування керованого потоку для SCADA-системи та бази даних. Практичне значення полягає у використанні отриманих результатів для порівняльного вибору сценаріїв обробки технологічних даних з урахуванням мережевого трафіку, стабільності затримки та ресурсів периферійного вузла.

Ключові слова: технологічні дані, потокова обробка, периферійна обробка, SCADA-система, редукація потоку.

DOI: 10.15587/1729-4061.2026.364578

ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ АВТОМАТИЗАЦІЇ ТРАНСПОРТУВАННЯ ГАЗУ НА ОСНОВІ СИСТЕМО-ТИПОЦЕНТРИЧНОГО МЕТОДУ (с. 111–138)

І. Б. Полатайко, Л. М. Заміховський

Об'єктом дослідження є процес архітектурного проектування системи автоматизації транспортування газу на рівні компресорної станції як мультиаспектної гетерогенної комп'ютерно-програмної системи в галузі автоматизації. Дана система поєднує підсистеми різної фундаментальної архітектурної природи – від PLC-програм до хмарних аналітичних систем та інформаційних систем з операційною логікою.

Дослідження сфокусовано на вирішенні проблеми повноти, релевантності та інтероперабельності артефактів архітектурного проектування складних комп'ютерно-програмних систем автоматизації. Існуючі методи зосереджені переважно на окремих аспектах – патернах проектування, інтеграції та рівності систем, оптимальному керуванні чи впровадженні певних можливостей, наприклад диспетчерського управління. Вони не формують єдиного наскрізного процесу архітектурного проектування системи в цілому.

Суть отриманих результатів полягає в формуванні архітектурної моделі системи шляхом проведення процесу архітектурного проектування із застосуванням системо-типоцентричного методу. Побудована архітектурна модель відрізняється простежуваністю архітектурних артефактів і вибором архітектурних точок зору залежно від фундаментальної природи систем, чим забезпечується покриття фундаментальних архітектурних аспектів та досягається повнота архітектурного опису. Її побудовано шляхом послідовного формування представлень системи з релевантних точок зору.

Результати демонструють науково-практичну доцільність методу для комплексного проектування складних гетерогенних систем автоматизації, що поєднують операційно-технологічний (ОТ) та інформаційно-технологічний (ІТ) рівні. З практичної точки зору, отримані результати можуть бути застосовані як референтна архітектурна модель для впровадження методу.

Ключові слова: архітектурне проектування, автоматизація, транспортування газу, системо-типоцентрична парадигма.

DOI: 10.15587/1729-4061.2026.359626

РОЗРОБЛЕННЯ АДАПТИВНО-ПРЕДИКТИВНОЇ СИСТЕМИ КЕРУВАННЯ ГІДРОДИНАМІЧНИМИ РЕЖИМАМИ ТРАНСПОРТУВАННЯ НАФТИ (с. 139–152)

О. В. Кучмистенко, М. В. Шавранський, Ю. Я. Пук

Об'єктом дослідження є процес автоматизованого керування роботою насосних станцій магістрального нафтопроводу під час пуску, зупинки та підтримання тисків при перекачуванні нафти. Збурення розглядаються як детерміновані параметричні варіації коефіцієнта гідравлічного опору, що залежать від зміни тисків, витрат, у тому числі в'язкості нафти. Розроблена система керування, що вирішує проблему інтеграції рекурсивної ідентифікації параметрів моделі з прогнозним формуванням керуючих впливів у єдиному контурі, який охоплює пуск, операційний режим та компенсацію збурень. Основна ідея автоматизації полягає у передачі функцій керування нафтопроводом від операторів насосних станцій до диспетчерського пункту.

Запропоновано адаптивно-предиктивну структуру керування, що поєднує рекурсивну ідентифікацію параметрів, оцінювання стану та оптимізаційне прогнозне формування керуючих впливів. Система інтегрує фільтр Калмана для оцінювання станів у реальному часі та модуль рекурсивного оновлення параметрів моделі. Це забезпечує онлайн-адаптацію математичної моделі об'єкта без втручання персоналу. Застосування запропонованого підходу зменшує накопичення інтегральної похибки регулювання витрати в операційному режимі приблизно на 87 відсотків. Після стрибка в'язкості витрата не виходить за межі технологічного допуску ± 2 відсотки. Покращення якості керування пояснюється здатністю системи прогнозувати динаміку об'єкта та компенсувати невив значеності до їх впливу на керовані змінні. Алгоритм автоматично відпрацьовує повний цикл: керований пуск, вихід на операційний режим та підтримання тисків у заданих межах. Результати можуть бути впроваджені в SCADA-системи нафтоперекачувальних станцій з керуванням з диспетчерського пункту. Це підвищить економічну ефективність та стабільність режимів транспортування нафти.

Ключові слова: фільтр Калмана, рекурсивна ідентифікація, керування гідродинамічними режимами, адаптивне керування, прогнозне керування, математична модель.