

Designing software for risk assessment using a Neural Network. Anna V. Lebedeva (lebedevaa28@gmail.com), Anna I. Guseva (AIGuseva@mephi.ru)



Anna V. Lebedeva



Anna I. Guseva

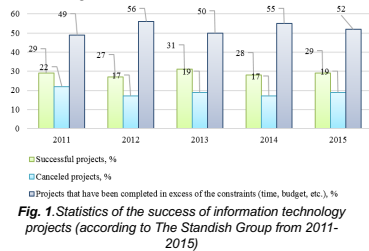


ABSTRACT

This report presents the results of the research in terms of using mathematical methods for the risk management process in the implementation of software development projects. Within the framework of this research, the risk assessment software was designed to assess the situation in the project, predict the future effectiveness of the project, and build scenarios to support decision-making. Such software will allow to combine all the actions of the analyst in one tool, where all the information about the project and the external environment will be stored, updated and constantly used for training the Neural Network apparatus on which the software is designed

INTRODUCTION

This work is devoted to solving an important scientific and practical problem of project management, namely, risk assessment and forecasting.



Based on the information received during the project, protective measures are developed to reduce the risk of implementation of risks. Within the framework of this research, software is being designed to support the risk assessment process in software development projects using the capabilities of Neural Network modeling.

To create a project risk assessment system using Neural Networks, there are several problems that need to be solved:

- availability of a dataset with time, technical, and financial indicators for a large number of IT projects
- classification of IT project risks
- reducing risks from qualitative to quantitative
- design and implement a specific network architecture for this task

METHODS

There are various methods of machine learning and data mining that can be applied to solve the problem of this research, for example:

- decision trees
- self-organizing Kohonen maps
- multi-layer Neural Networks

In this study uses multi-layer Neural Networks.

The design and training of a neural network for risk recognition in software development projects was carried out using the Deductor Studio Lite tool. Fig. 2 illustrates Neural network training.

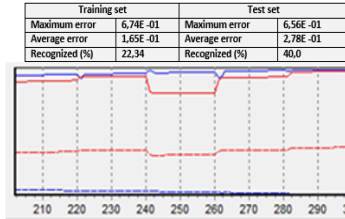


Fig. 2. Neural network training

ENTER FINDING #1

Setting a function for getting a set of key project indicators:

$$F_i = (k_i, z_i) = F_i(u_i, z_{i-1}, p_i)$$

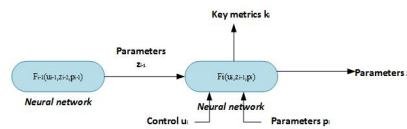


Fig. 3. Scheme of the Neural Network model

It should also be noted that for more accurate forecasting of project risks, it is necessary to work on their formalization (in terms of sources of risk, influencing factors and countermeasures).

ENTER FINDING #2

As part of previous research, we have already prepared a dataset, risk classification of an IT project with the development of a cognitive map for risk assessment.

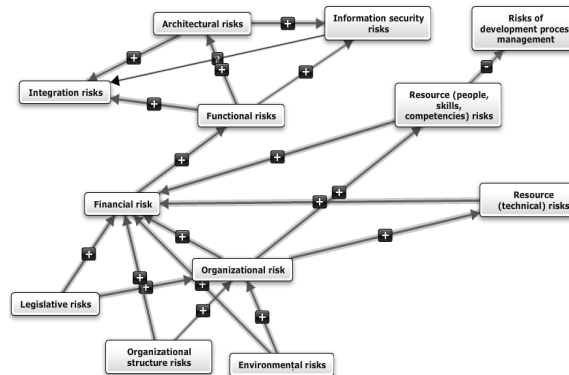


Fig. 4. Cognitive map of risk factor groups

ENTER FINDING #3

As a result of building the network, a multi-layer network (16x7x1) is obtained, that is, a network that consists of 16 input neurons 7 neurons of the hidden layer and 1 neuron in the output layer.

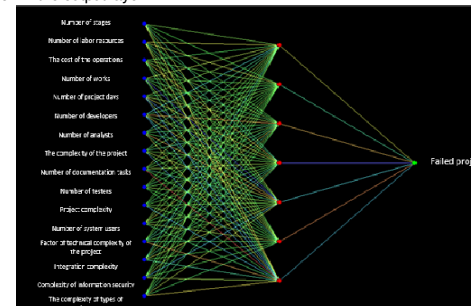


Fig. 5. 16x7x1 Neural Network

DISCUSSION

The study showed that neural networks are better used with a large enough amount of initial data, otherwise the time, financial and technical costs will be inadequate compared to the output data. It is desirable to structure all data in a form suitable for the neural network architecture. Thus, all risk factors should be classified and ranked according to their impact and probability. For some architectures, the impact or probability will be evaluated by the Neural Network itself.

During the use of this approach, 20 projects of various scale and type were evaluated, including projects for the development of technical software documentation, for the development of concepts of information systems, for the development of information systems, for the finalization of ready-made solutions.

The most accurate and significant results were achieved in the framework of projects for the development of reporting documentation.

It should also be noted that the complexity of the project manager involved in the calculation of risks and cost and complexity, decreased by 3 times. Thus, the project manager can take part in a much larger number of pre-sale works and carry out a more rapid calculation for the provision of commercial proposals to potential customers.

CONCLUSIONS

1. Developed approach has already shown positive results after implementation at the enterprise. The time spent by the project manager to evaluate the project decreased by 3 times
2. Neural networks are better used with a large enough amount of initial data
3. All risk factors should be classified and ranked according to their impact and probability

ACKNOWLEDGMENTS & REFERENCES

This work was supported by Russian Foundation for Basic Research (RFBR) grant № 20-010-00708/20.

References

- Golovko V. A. Neural networks: training, organization and application. IPRZhR, Moscow, 2001.
- Carbonell J. G., Michalewski R. S., Mitchell T. M. Machine Learning. Symbolic Computation. Springer, Berlin, Heidelberg, 1983.
- Korneev D. S. (2007). Using neural networks to create a model for evaluating and managing enterprise risks. Managing large systems: proceedings, 17, 81-102.
- Chandan Kumar, Dilip Kumar Yadav (2015). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. Procedia Computer Science, 54, 353-361, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2015.06.041>.
- Samsonov A. V. (2018). Schema formalism for the common model of cognition. Biologically Inspired Cognitive Architectures, 26, 1-19.
- Lebedeva A.V., Guseva A.I. (2020). Cognitive maps for risk estimation in software development projects. Advances in Intelligent Systems and Computing, 948, 295-304.
- Ilicheva A. N., Korovin D. I. (2007). Neural Network modeling capabilities for improving management accounting in the intra-company budgeting system. Modern science-intensive technologies. Regional Appendix, 2-3, 22-27.