UDC 004.8; 005.33
DOI: 10.15587/1729-4061.2025.322051

IMPROVING A RISK ESTIMATING METHOD FOR THE "SMART HOUSE" INFORMATION SYSTEM IT PROJECT

Maksym Ievlanov
Corresponding author
Doctor of Technical Sciences, Professor*
E-mail: maksym.ievlanov@nure.ua
Nataliia Vasyltsova
PhD, Associate Professor*
Iryna Panforova
PhD, Associate Professor*
Yevhen Kliuvanskyi
Student*
*Department of Information Control System
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166

The object of this study is the IT project risk management process.

The study solves the task to improve the accuracy of risk assessment in IT projects and, in particular, IT projects to design the information system (IS) "Smart House". Research into this area is mainly focused on the application of machine learning (ML) methods to improve the results of conventional evaluation methods. Issues related to quantitative risk assessment of IT projects to design the "Smart House" IS remain practically unexplored.

During the study, it was proposed to use ML methods for preprocessing the raw data. For this purpose, a combined risk assessment method was devised. In this technique, methods of Support Vector Machine and Bayesian networks were applied to process the raw data. The results of their application were used as input data for Monte Carlo simulations.

During the software implementation of the devised method, its technological stack was determined. Fragments of the program code are given, which describe the implementation of the basic elements of the combined method.

The devised method and its software implementation were used to assess the risk of delay in the implementation of the IT project to design the "Smart House" IS. The evaluation results determine the expected duration of this IT project at 234.5 days, with a deviation range of 226–244 days (with a 95 % confidence interval). The results of a comparative analysis of the obtained estimates with estimates of the same risk obtained using the conventional Monte Carlo method show that the devised method provides higher reliability of forecasts.

The application of research findings makes it possible to improve the quality of managing IT projects by increasing the accuracy and reliability of their risk assessments

Keywords: smart house, IT project, risk assessment, Monte Carlo method, Support Vector Machine, Bayesian network

Received 09.10.2024
Received in revised form 29.11.2024
Accepted 16.12.2024
Published 21.02.2025

How to Cite: *Ievlanov, M., Vasyltsova, N., Panforova, I., Kljuvanskij, Y. (2025). Improving a risk estimating method for the "Smart house" information system IT project. Eastern-European Journal of Enterprise Technologies, 1 (3 (133)), 6–16. https://doi.org/10.15587/1729-4061.2025.322051*

1. Introduction

Given the rapid development of information technologies (IT) and the growing interest in intelligent automation of living spaces, IT projects to design information systems (ISs) "Smart House" are becoming especially relevant. The class of ISs, which is called "Smart house", includes complex integrated automation systems designed to control and manage various engineering systems in a modern building [1]. Such engineering systems may include automatic control of lighting, ventilation, heating, security, and even household appliances [2].

However, along with the numerous advantages offered by Smart Home ISs, they also face a number of challenges and risks. IT project risk means the potential occurrence of an event or series of events that may negatively affect the success of the IT project. For IT projects to design the "Smart House" ISs, risk assessment becomes a critically important task [3]. The complexity of solving this problem is due to a number of determining factors related to technical, commercial, and organizational aspects. In addition to these factors, it should be taken into account that existing approaches mainly focus on general aspects of risk management. At the same time, the features that are specific to IT projects to design the "Smart House" IS are almost not taken into account. In our opinion, such features should include:

– high competition in the market of "Smart House" ISs and services for the development, implementation, and maintenance of similar systems;

– the customer has an idea of the cost of the "Smart House" IS and the functions it will be able to perform, even before the formation of the technical task for this system;

– the need to maintain a high level of integration of various ITs and platforms;

– the need to ensure a high level of system security and reliability;

– high dynamism of changes that occur during the planning and execution of an IT project.

The main directions in devising modern approaches, methods, and technologies of project management in this segment of the IT industry are optimization of development, reduction of risks, and improvement of the efficiency of project implementation [4]. Based on this, conducting research on improving the efficiency and quality of risk management of IT projects to design the "Smart House" ISs is relevant from both a theoretical and a practical point of view.

2. Literature review and problem statement

Risk assessment refers to the process of identifying and analyzing events, as well as responding to them. At the same time, the goal is to maximize the probability of favorable events and their consequences and to minimize the probability and consequences of unfavorable ones. However, quite often they are limited only to work with adverse events [5].

In the IT industry, there are many methods that are recommended for assessing risks in IT projects and, in particular, in IT projects to design the "Smart House" ISs. Each of these methods has specific advantages and limitations. The most common of such methods are the following [6]:

a) SWOT analysis;
b) PEST analysis;
c) method of expert evaluations;
d) Monte Carlo method;
e) sensitivity analysis method;
f) decision tree method.

But the application of each of these methods for risk assessment of such a variety of IT projects as IT projects to design the "Smart House" ISs is associated with significant problems. Thus, SWOT analysis and PEST analysis are too general and do not provide an in-depth analysis of the specific technical risks that are inherent in the "Smart House" ISs. They may miss the critical technical challenges of integrating different technologies and platforms. The limitations of these methods are that both analyses do not provide a quantitative assessment of risks, which complicates the decision-making process regarding the allocation of resources to minimize or avoid risks [7].

Among the known limitations of the method of expert evaluations, especially worth highlighting is its possible ineffectiveness in the case of high complexity of the project and a large number of integrated systems and components. These features are characteristic of the "Smart House" ISs [8].

The Monte Carlo method requires large computing resources and high skill of the analyst for proper modeling. These requirements may not be achievable for small teams or projects with limited budgets. In addition, the Monte Carlo method may not take into account new or non-standard risks that quickly appear in the dynamic technological landscape of the "Smart House" ISs [9].

To use sensitivity analysis effectively, probability distributions for all key variables must be accurately determined. This could be difficult without sufficient historical data or a deep understanding of project processes. In addition, the use of low-quality or outdated data could lead to incorrect risk assessments.

Applying the decision tree method to assess the risks of complex ISs "Smart House" may become an extremely complicated procedure that requires too much time. Also, this method may not take into account the influence of external factors that may change suddenly. The use of decision trees could be limited in projects with a high degree of uncertainty and innovation, where it is difficult to predict all possible scenarios [8].

It should be noted that in the mid-2010s there have been changes in the description of life cycle processes of systems and, in particular, IT products. One of the results of these changes is the creation of a process of system analysis [10], which is aimed at applying a formal apparatus of system analysis to solve individual problems within the life cycle of an IT product. One of the varieties of such tasks is the task of evaluation. The application of the process of system analysis makes it possible to consider different assessment tasks within the life cycle of an IT product as variants of a typical assessment task, which is solved using a limited set of methods.

Such a view allows us to make an assumption about the possibility of using the same (or similar) methods for solving such evaluation problems as the problems of evaluating time, cost, effort, and risks in IT projects. It was this assumption that formed the basis of modern research in the field of IT project management.

No less important is the assumption about the expediency of using modern models and methods of artificial intelligence to solve assessment problems in IT projects. In [11], the results of a systematic review of the scientific literature on solving the problems of estimating the effort and cost of software development are reported. It has been established that the entire set of methods for solving these problems could be divided into two classes: methods based on machine learning (ML methods) and methods that do not use machine learning (nonML methods). Neural networks are the leader among first-class methods in terms of the number of dedicated studies. Ensemble and regression methods took the second and third places, respectively, in the first class of methods with a slight lag [11].

Among the methods of the second class, the COCOMO model takes the first place in terms of the number of dedicated studies by a large margin. The second place is occupied by the method of functional points (Functional Point Analysis). The third place was shared by all other methods that are not based on machine learning [11].

The analysis results Published 21.02.2025in [11] allow us to draw the following conclusions:

– the COCOMO parametric evaluation model is recognized by the scientific community as the most recommended model for use when solving the tasks of estimating the effort and cost of IT projects;

– there is no clear leader among ML methods that would be recommended in the vast majority of scientific research.

Current studies [12, 13] address the peculiarities of the use of neural networks for estimating the effort and cost of IT projects of various purposes. The main reason for the use of neural networks is the inaccuracy of decisions obtained using existing evaluation methods [13]. In addition, it is recognized that the existing methods of assessment are quite strongly dependent on the individual characteristics of human thinking [12]. But the neural networks proposed in these publications are of value mainly to scientists. In [12] it is recognized that the solution to the problem of applied usefulness of ML-methods and options for their further development requires further research.

Similar trends are observed in other types of projects. Thus, modern research in the field of construction project management suggests using general regression neural networks to assess the risk of construction delays [14], and an innovative approach that combines binary programming with genetic algorithms to assess risks and their impact on the budget of a construction project [15]. It should also be noted that the neural networks proposed in [14] were compared with such risk assessment and prediction methods as support vector machines and Tree Boost. However, comparisons of productivity and accuracy of neural networks with existing risk assessment methods were not carried out in [14]. Therefore, the question of the practical feasibility of using ML-methods for solving evaluation problems in project management remains open.

As for solving the problems of risk assessment in IT projects, the search for more accurate and practical methods of solving these problems continues. For example, in [16] it is proposed to use risk prediction trees and Bayesian network methods to assess risks in IT projects at the stage of software development. In [17], the problem of risk management during software development was formally described as a problem of linear programming and its solution was proposed, which makes it possible to minimize the costs of possible risk losses. But none of these results were compared with ML methods and nonML methods simultaneously.

The same trend is present in studies on solving the issue of risk assessment in IT projects to design the "Smart House" ISs. But the need for practical application of research results significantly limits the choice of promising risk assessment methods. For example, in [18] it is proposed to use the FMEA method and empirical analysis tested in practice to assess the risks of including intelligent robotics technologies in the internal logistics systems of "smart houses". In general, the issue of risk assessment in IT projects to design the "Smart House" ISs is associated in modern research mainly with issues of safety and security of Internet of Things devices as important components of the system against hacking and unauthorized access [19]. At the same time, security issues are considered not only from a technical but also from a financial perspective (ensuring the security of financial transactions in the process of re-ordering products with a "smart" refrigerator) [20]. Issues of assessing other groups of risks in IT projects to design the "Smart House" ISs are almost not considered in modern research.

Based on the results of our review of modern research, the following conclusions are drawn:

a) the use of widely used risk assessment methods in IT projects to design the "Smart House" ISs is limited by the shortcomings of these methods;

b) modern research does not make it possible to single out an objective leader from the set of ML-methods, the application of which to solve evaluation problems in IT projects would ensure an increase in accuracy and would be expedient from an applied point of view;

c) modern research in the field of risk management of IT projects for the development of the "Smart House" ISs is mainly focused on the application of conventional methods for managing the safety and security of Internet of Things devices.

These conclusions, based on the results of modern research, are the consequences of the following reasons:

a) a relatively small number of studies in the field of quantitative risk assessment of IT projects and the use of modern ML models and methods for such assessment;

b) the need to pay the main attention in IT projects to design the "Smart House" ISs to applied research related to risk assessment of the most vulnerable part of the system;

c) lack of available datasets that would allow for a comparative analysis of new or improved and conventional risk assessment methods;

d) unique features of each IT project and difficulties that arise during the generalization and formalization of their descriptions, current and final results.

As promising areas of research into models and methods for solving the assessment problem in the field of IT project management, the following are highlighted in [11]:

a) obtaining models and methods adapted to the analysis of complex projects (ML models are prone to estimation errors; determining the correct ML models and their application in software evaluation remains a problem);

b) development of benchmarks for evaluation (non-acceptance of benchmarks is a problem that needs to be mitigated in order to determine a course of action for future research and obtain a tool for objectively determining the effectiveness and scalability of models and evaluation methods);

c) elimination of uncertainty of handling (software development by its nature remains uncertain, which must be taken into account in cost estimation models and methods);

d) data availability (the availability of multi-functional, complex data sets is a challenge, as many IT companies keep the data undisclosed for privacy reasons);

e) the quality of research methodology (researchers must find a balance between the importance of a clear reporting methodology and the possibility of repeated studies of the reported results).

Based on the promising areas of research considered in [11], the task of our study is formulated as the development of methods adapted to the assessment of risks within a complex IT project to design the "Smart House" IS. Solving this task could significantly increase the efficiency of resolving the assessment problem and the accuracy of the results through a slight increase in the complexity of solving this task.

3. The aim and objectives of the study

The purpose of our study is to improve the Monte Carlo method for risk assessment of the IT project to design "Smart House" systems. This improvement makes it possible to increase the accuracy of solving the problem of IT project risk assessment.

To achieve the goal, the following tasks were set:

– to devise a combined risk assessment method using ML methods and the Monte Carlo method;

– to determine the features of the applied implementation of the devised method;

– to conduct an experimental test of the devised risk assessment method for the "Smart House" IS design project.

4. The study materials and methods

The object of our study is the process of risk management as one of the processes in the technical management of an IT project [10]. The subject of the study is methods of risk assessment in IT projects to design the "Smart House" systems.

The main hypothesis of the study assumes that the use of ML methods not to improve the accuracy of already obtained risk assessments but to improve the initial data would make it possible to eliminate the shortcomings of existing risk assessment methods.

As an example of existing risk assessment methods, it is proposed to use the Monte Carlo method. It encompasses the systematic process of creating random samples to simulate various scenarios and evaluate the potential outcomes of a project or process. This method consists of the following sequence of steps [9]:

Step 1. Determination of input variables affecting project results. These variables may include the duration of tasks, availability of resources, market conditions, and other factors that may introduce uncertainty.

Step 2. For each input variable, probability distributions representing their possible values and associated probabilities are determined. Common distribution types used in Monte Carlo simulations include the normal, uniform, triangular, and log-normal distributions.

Step 3. Random samples are generated for each input variable based on their probability distribution. This involves extracting values from a distribution using appropriate random number generation techniques such as back-transformation or accept-reject methods.

Step 4. After generating random samples for all input variables, calculations or simulations are performed using these values to determine project outcomes. This may include running computer models, performing mathematical calculations, or performing simulations specific to the subject area of the project.

Step 5. Monte Carlo simulations are usually repeated many times to generate a statistically significant number of scenarios. Each iteration of the simulation involves generating a new set of random samples for the input variables and running the computation or simulation again.

Step 6. The results of each simulation iteration are collected and analyzed to determine the range of possible outcomes. Statistical analysis methods such as mean, standard deviation, percentiles, and confidence intervals are used to generalize and interpret simulation results.

Step 7. The distributions created from the simulation results give an idea of the probability of different outcomes. Based on these analyses, the risk and uncertainty associated with the project could be assessed.

Step 8. Based on the information received, project teams could refine and optimize their plans, resource allocation, and risk management strategies. This process enables continuous improvement.

Two criteria are used to quantify the results: average value and uncertainty [9]. The average value is one of the main statistical indicators that makes it possible to get an idea of the central tendency of the simulation results, determined from the following formula:

$$M = \frac{1}{N}\sum_{n=1}^{N} f_n, \tag{1}$$

where M is the average value; N is the number of experiments; f_n – obtained indicator values.

Uncertainty (coefficient of variation) is calculated from the following formula:

$$\varepsilon = \frac{1}{M}\sqrt{\frac{\sum_{n=1}^{N}\left(f_n - M\right)^2}{N}}, \tag{2}$$

where ε – uncertainty, the dispersion characteristic of the obtained values; f_n – received values of the indicator; M is the expected value of the random variable f_n; N – number of experiments.

The uncertainty shows the degree of variation of the simulation results around the mean value. The smaller the coefficient of variation, the more accurately the average value characterizes the expected value. Uncertainty could also be considered as an assessment of the risk associated with the fact that the obtained value of the indicator will deviate from the expected value of a random variable [9].

The use of the Monte Carlo method for risk assessment provides the following advantages: objective risk assessment; consideration of uncertainty; visualization of evaluation results; identification of critical factors. In general, the method makes it possible to quantitatively assess the risks associated with delays in the schedule, exceeding the budget and other key indicators of the project, which is extremely important for complex and multi-component projects. Due to the use of probability distributions for the input variables, the Monte Carlo method makes it possible to take into account the uncertainty and variability of project parameters, which is especially relevant when working with new or non-standard technologies [21].

As the main disadvantages of the Monte Carlo method, the significant costs of computing resources and the underestimation of new or non-standard risks were indicated above. Additional disadvantages of this method are complexity in data preparation; risk of misinterpretation; dependence on the quality of input data (poor or outdated data could lead to incorrect risk assessments).

5. Results of devising an ensemble of ML-methods and the Monte Carlo method for risk assessment of an IT project to design "Smart House" systems

5. 1. Results of devising the combined risk assessment method

To test the main hypothesis of our study, a combined method of risk assessment was devised, which consists of the following stages and steps:

Stage 1. Collection and preparation of data for risk assessment.

Stage 2. Construction of machine learning models.

Stage 3. Monte Carlo simulations.

Using machine learning models to determine the interdependence between the task execution time and its characteristics, simulation using the Monte Carlo method evaluates the impact of possible delays in each task on the final implementation of the project [22]. Analysis of simulation results reveals critical tasks that have a significant impact on project completion. This makes it possible to devise targeted risk reduction strategies, focusing on the most influential tasks and their risks [23].

Stage 1 consists of the following steps:

Step 1. 1. Determination of data requirements.

Step 1. 2. Data collection.

Step 1. 3. Data cleaning.

Step 1. 4. Data transformation.

Step 1. 5. Data integration.

Step 1. 6. Construction of training and testing sets.

Let's consider the content of these steps in more detail.

Defining data requirements (Step 1. 1) consists of defining key variables and identifying data sources. Variables that affect risk in an IT project shall be called key. For a "Smart House" IS, such variables may include the cost of components, installation time, supplier reliability, technology compatibility, etc. Potential data sources include project reports, financial records, supplier performance data, historical project databases, and stakeholder interviews [24].

In Step 1. 2, both structured data (for example, numerical data in databases) and unstructured data (for example, text in project reports) are collected. External data (market trends, technological progress, regulatory changes, etc.) that may affect the project are also collected.

At Step 1. 3, a check is performed for the presence of missing data and, depending on the volume and nature of the missing data, work is carried out to remove or replace them. Examples of missing data are empty values of indicators needed for further analysis or missing values of variables in datasets. As a procedure for replacing missing data, one could use, for example, substitution of the average value.

In Step 1. 4, normalization or standardization techniques are applied to scale the data, making it suitable for machine learning algorithms that are sensitive to the scale of the input data. Data normalization is performed, based on [25], using the formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{3}$$

where x' is the normalized value; x is the initial value; min(x) is the minimum value in the data set; max(x) is the maximum value in the data set. Data standardization, based on [25], is performed according to the formula:

$$z = \frac{x - \mu}{\sigma}, \tag{4}$$

where z is a standardized value; x is the initial value; μ is the average value in the data set; σ is the standard deviation in the data set. This step also involves converting the categorical data into a numeric format, making it interpretable for machine learning models. For this conversion, it is proposed to use the Label Encoding and One-Hot Encoding methods [26].

Step 1. 5 integrates data from different sources to build a comprehensive dataset. This step may include:

– data alignment on a common timeline;

– combining data sets based on common identifiers;

– aggregation of data points to the appropriate level of detail;

– derivation of new variables that could better reflect the dynamics of project risks.

In Step 1. 6, the data is divided into training and test sets. It is proposed to use 70 % of the data for training and 30 % of the data for testing. The data are shuffled randomly to ensure an even distribution of examples in both samples.

Completing the steps of Stage 1 ensures that the data used for analysis in the combined risk assessment method are reliable and structured in a way that maximizes the effectiveness of the subsequent modeling and simulation processes.

Stage 2 consists of the following steps:

Step 2. 1. Selection of features.

Step 2. 2. Model training.

Step 2. 3. Model validation.

Let's consider the content of these steps in more detail.

Feature selection, performed in Step 2. 1, is the process of selecting from the available data set the most relevant attributes (characteristics) that make a significant contribution to predicting outcomes. The purpose of feature selection is to improve model performance, reduce overtraining, enhance generalization, and speed up training processes. Effective feature selection could also help understand what factors affect the target variable (the variable that the machine learning model is trying to predict or explain). To select and assess the importance of features in Step 2. 1, it is suggested to use the correlation coefficient or the chi-square test (in the case of categorical variables).

To train the model (Step 2. 2) in the combined method of risk assessment, it is proposed to use such ML methods as Support Vector Machines (SVM) and Bayesian networks.

SVMs aim to maximize the difference between different classes in a dataset, which helps clearly define decision boundaries. This is particularly useful in risk classification tasks where it is important to accurately distinguish between high and low risk scenarios. SVM could handle non-linear input spaces by transforming the input data into a higher dimensional space where a linear separator could be found. Such flexibility makes it possible for SVM to model complex relationships between risk factors in IT projects. Learning from historical data, SVM predicts the probability of risk events, helping proactively manage potential problems in the implementation of the IT project to design "Smart House" ISs. By quantifying the probabilistic effect of one variable on another, Bayesian networks help determine how changes in one aspect of a project could spread to other areas.

Step 2. 2 of the model is iterative and may require several rounds of tuning and evaluation.

At Step 2. 3, the expected capability of the analytical model built on the basis of machine learning is checked, as well as the compliance with the requirements of the task being solved. Model validation is performed on an independent (i.e., not used for training and testing) validation data set after model training. Since in Step 2. 2 it was proposed to use the SVM method, in Step 2. 3 it is proposed to use such performance indicators as Mean Squared Error (MSE), Mean Absolute Error (MAE), coefficient of determination. These indicators make it possible to evaluate the accuracy and reliability of the model, as well as to compare its performance with other models.

The proposed steps of Stage 2 guarantee the improvement of the original data through the use of SVM and Bayesian networks. Such an improvement, in turn, is designed to increase the accuracy of Monte Carlo modeling of the predictive values of risks of IT projects to design the "Smart House" IS.

Stage 3 consists of the following steps:

Step 3. 1. Setting the simulation parameters.

Step 3. 2. Integration of machine learning predictions.

Step 3. 3. Starting the simulation.

Step 3. 4. Analysis and aggregation of results.

Step 3. 5. Visualization and reporting.

Let's consider the content of these steps in more detail.

In Step 3. 1, you need to determine the number of simulation runs of the Monte Carlo method (increasing the number of runs could lead to more accurate and stable estimates at the cost of increased computation time). Based on feature selection and model output, initial conditions and input variables are adjusted for each simulation run. This includes customizing probability distributions for each input that have been refined using machine learning predictions.

In Step 3. 2, the results of the application of ML methods (risk probabilities, expected delays, etc.) are determined as inputs or modifiers in Monte Carlo simulations.

At Step 3. 3, Monte Carlo simulation is started, which involves random sampling from probability distributions of input data for each iteration and calculation of the obtained results. For each simulation run, the results, such as the total cost and time to complete the project, are recorded. These results are used for further analysis and risk assessment, which makes it possible to determine the most likely scenarios of the development of events [27]. The estimate of project duration using the Monte Carlo method is determined, according to [28], from the following formula:

$$T_j = \sum_{i=1}^{n} T_{i,j}, \quad (5)$$

where T_j is the total duration of the project for simulation; $T_{i,j}$ is the duration of task i in simulation j, which is randomly generated from the probability distribution for the task; n is the number of tasks.

Step 3. 4 analyzes and aggregates the simulation results to identify trends, probabilities of different outcomes, and potential critical risk factors. Statistics such as mean, median, variance, and confidence intervals for the results are calculated.

Step 3. 5 constructs histograms, scatter plots, and box plots that illustrate the distribution of simulation results. This representation of results helps visually assess risks and their consequences [29]. Also, at this step, the obtained results are combined into a comprehensive report, which provides the reader with an idea of the most significant risks of the IT project, their likely consequences, and recommended mitigation strategies.

The result of the application of the devised combined method of IT project risk assessment is the assessment of time risks of project delays. This estimate is formed on the basis of the application of the results of the analysis of the historical data of the schedule of previous projects to further forecast the possible completion time of the IT project and its individual tasks.

5. 2. Results of applied implementation of the combined risk assessment method

To implement the devised combined method of risk assessment, it was proposed to develop specialized software. For this product, a set of technologies has been formed that are used together to develop and support the software of the service. Such a set of technologies will be referred to hereafter as a technological stack [30].

The technological stack of the software implementation of the combined method of risk assessment consists of the following elements:

a) Python programming language (it is the main programming language that was used to devise machine learning models and conduct Monte Carlo simulations);

b) Google Colab cloud service for executing Python codes (makes it possible to use free computing resources, supports collaborative work, and makes it possible to save work in Google Drive);

c) Scikit-learn machine learning library (contains a wide range of algorithms for classification, regression, and clustering, as well as tools for data preparation, model selection, and assessment of its quality);

d) pgmpy machine learning library (Python library for building and working with probabilistic graphical models, including Bayesian networks);

e) Pandas data processing library (Python library designed for working with structured data);

f) NumPy data processing library (Python library for working with large multidimensional arrays and matrices, as well as for performing operations on these arrays);

g) visualization tools Matplotlib and Seaborn (Python libraries for drawing graphs, charts, and other means for visualizing simulation results).

A fragment of the code listing for the preparation of historical data, training of the SVM model, execution of forecasting and model evaluation tasks is shown in Fig. 1. A fragment of the code listing for the preparation of project data, forecasting the risks of delays, and constructing a graph of project tasks is shown in Fig. 2. A fragment of the Monte Carlo simulation code listing and visualization of its results is shown in Fig. 3.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Loading historical data
historical_data_path = '/content/historical_data.csv'
historical_data = pd.read_csv(historical_data_path)

# Preparing data for training an SVM model
X_hist = historical_data[['Duration', 'Cost', 'Dependency']]
y_hist = historical_data['Delay_Risk']

# Splitting data into training and testing sets
X_train_hist, X_test_hist, y_train_hist, y_test_hist = train_test_split( X_hist, y_hist, test_size=0.2, random_state=42 )

# Scaling data
scaler_hist = StandardScaler()
X_train_hist_scaled = scaler_hist.fit_transform(X_train_hist)
X_test_hist_scaled = scaler_hist.transform(X_test_hist)

# Training an SVM model
svm_model_hist = SVR(kernel='linear')
svm_model_hist.fit(X_train_hist_scaled, y_train_hist)

# Predicting and evaluating the model
y_pred_svm_hist = svm_model_hist.predict(X_test_hist_scaled)
mse_svm_hist = mean_squared_error(y_test_hist, y_pred_svm_hist)
mae_svm_hist = mean_absolute_error(y_test_hist, y_pred_svm_hist)
r2_svm_hist = r2_score(y_test_hist, y_pred_svm_hist)

print(f'Mean Squared Error for SVM (Historical Data): {mse_svm_hist}')
print(f'Mean Absolute Error for SVM (Historical Data): {mae_svm_hist}')
print(f'R^2 for SVM (Historical Data): {r2_svm_hist}')
```

Fig. 1. A fragment of the code listing for historical data preparation, SVM model training, forecasting, and model evaluation tasks

```
# Loading new project data
project_data_path = '/content/smart_home_project_tasks.csv'
project_data = pd.read_csv(project_data_path)

# Scaling project data
X_new_project = project_data[['Duration', 'Cost', 'Dependency']]
X_new_project_scaled = scaler_hist.transform(X_new_project)

# Predicting delay risks for a new project
predicted_risks_new_project = svm_model_hist.predict(X_new_project_scaled)

# Creating a project task graph
G = nx.DiGraph()
for i, row in project_data.iterrows():
    G.add_node(row['Task_ID'], duration=row['Duration'], cost=row['Cost'], dependency=row['Dependency'])
    if row['Dependency'] != 0:
        G.add_edge(row['Dependency'], row['Task_ID'])

# Visualizing the critical path
plt.figure(figsize=(12, 8))
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_size=3000, node_color='skyblue', font_size=10, font_weight='bold', arrowsize=20)
plt.title('Project task graph')
plt.show()
```

Fig. 2. Listing of the project data preparation code, forecasting the risks of delays, and constructing a graph of project tasks

```
# Conducting Monte Carlo simulations to estimate project duration
n_simulations = 1000
simulated_durations_new = []

for _ in range(n_simulations):
    total_duration = 0
    for i, row in project_data.iterrows():
        predicted_risk = predicted_risks_new_project[i]
        # Applying delay with random deviation
        duration_with_delay = row['Duration'] * (1 + np.random.normal(predicted_risk, 0.1))
        total_duration += duration_with_delay
    simulated_durations_new.append(total_duration)

# Analyzing simulations results
simulated_durations_new = np.array(simulated_durations_new)
mean_duration_new = np.mean(simulated_durations_new)
confidence_interval_new = np.percentile(simulated_durations_new, [2.5, 97.5])

print(f'Simulated Mean Duration (New Project): {mean_duration_new}')
print(f'95% Confidence Interval (New Project): {confidence_interval_new}')

# Visualizing simulations results
plt.hist(simulated_durations_new, bins=20, edgecolor='k', alpha=0.7)
plt.axvline(mean_duration_new, color='r', linestyle='dashed', linewidth=2, label=f'Mean Duration: {mean_duration_new:.2f} days')
plt.axvline(confidence_interval_new[0], color='g', linestyle='dashed', linewidth=2,
               label=f'95% Confidence Interval: {confidence_interval_new[0]:.2f} - {confidence_interval_new[1]:.2f} days')
plt.axvline(confidence_interval_new[1], color='g', linestyle='dashed', linewidth=2)
plt.title('Distribution of Project Duration Based on Monte Carlo Simulations')
plt.xlabel('Project Duration (days)')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

Fig. 3. Code listing for Monte Carlo simulations and visualization of their results

5. 3. Results of experimental verification of the devised combined risk assessment method

The "Smart House" IS design project includes the development of an integrated system that makes it possible for users to remotely control home systems using an intuitive interface of a mobile application or a web interface. The devised "Smart House" system includes modules for lighting, heating, security, and access control, but could be expanded with other modules (modules for power supply management, interactive systems, water supply, animal and plant care, emergency control, etc.).

The description of tasks within the IT project to design a "Smart House" IS is given in Table 1. In particular, this table contains information about the projected duration of each task, its cost, and interdependences between tasks (the number of the task that immediately precedes the task described in a given line of the table).

Modeling of an IT project to design the "Smart House" IS was carried out in two ways. First, the devised combined risk assessment method was applied. During Stage 2 of this method, a graph of IT project tasks was formed with the help of the developed application implementation of the method, presented in Fig. 4. In this figure, IT project tasks that are on the critical path are highlighted in red.

To assess the stability of the model obtained from the results of Stage 2 of the combined method, cross-validation was carried out using 5-fold cross-testing. This method makes it possible to check how well the model generalizes the data and avoids overtraining. Cross-validation results showed consistently low error values on different data subsets, which confirms the reliability of our model.

The model was tested on a separate test dataset that was not used during training. The MSE value on the test set was also low (0.00957), which confirms the high accuracy of our model predictions.

Analysis of the residuals (differences between actual and predicted values) revealed that the residuals were randomly distributed with no significant systematic deviations. This indicates that the model captures the underlying patterns in the data well and is not prone to overtraining.

Table 1

List and description of tasks within an IT project to design the "Smart House" information system

Task number	Task ID	Description	Responsible	Dura-tion	Cost	Interde-pendence
1	Analysis of user requirements	Study of user needs, formation of requirements for the system	Business analyst	10	1,000	0
2	Compiling a technical task	Development of specifications based on user requirements	Business analyst, Architect	12	1,500	1
3	Development of system archi-tecture	System architecture design	Architect	8	900	1
4	Development of databases	Devising a database schema	DBA, Developer	15	2,000	2
5	Creating an API	API development for interaction with the data-base and external systems	Developer	7	800	2
6	Development of a mobile appli-cation	Development of a mobile application for system management	Mobile developer	9	1,100	3
7	Web interface development	Construction of a web interface for system management	Web developer	13	1,300	4
8	Integration with management systems	Integration with existing smart home manage-ment systems	Integrator	11	1,200	5
9	Development of a lighting con-trol module	Building a module for lighting control	Developer	6	700	5
10	Development of a heating con-trol module	Building a module for heating control	Developer	14	1,600	6
11	Security module development	Building a module for security management	Developer	10	1,000	7
12	Development of the access con-trol module	Building a module to control access to the house	Developer	9	950	8
13	Functionality testing	Testing of all modules for correct operation	Tester	7	850	9
14	Integration testing	Integration testing between all modules	Tester	12	1,400	10
15	Security testing	Conducting system security testing	Security tester	8	900	11
16	Performance optimization	System optimization to ensure high perfor-mance	System engineer	13	1,500	12
17	User documentation	Preparation of documentation for end users	Technical writer	9	950	13
18	Pilot implementation	Pilot implementation of the system in selected houses	Integrator	6	700	14
19	User training	Training for system users	Coach	11	1,200	15
20	Final implementation	Final implementation of the system and provi-sion of support	Integrator	14	1,600	16

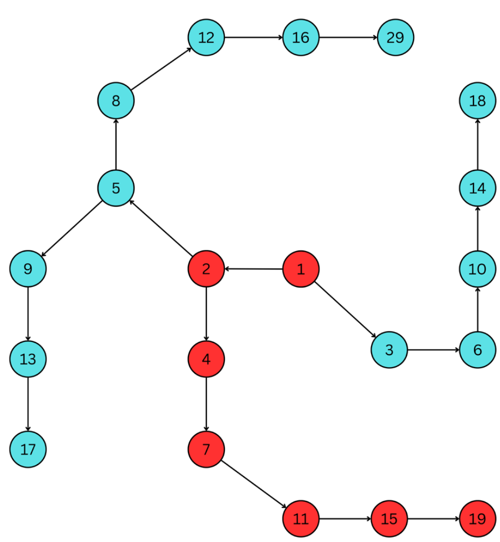
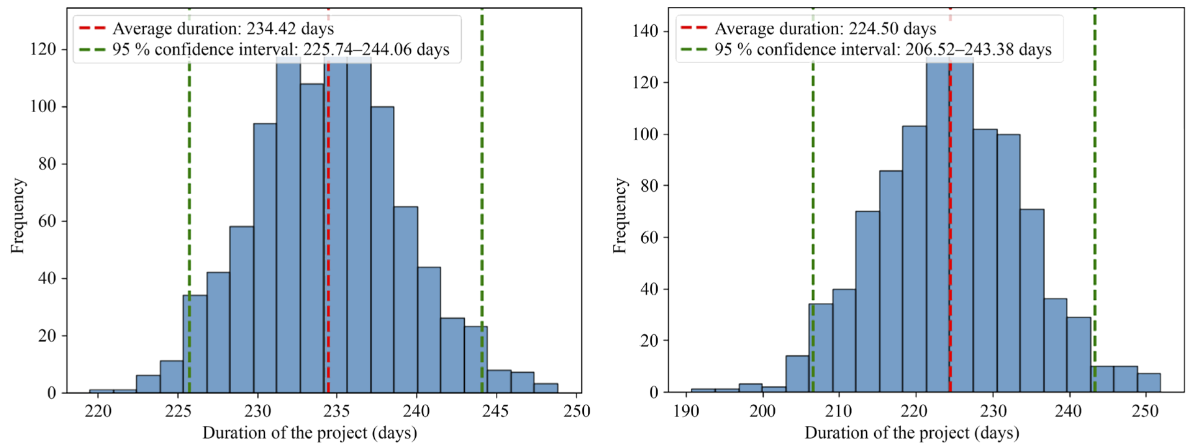


Fig. 4. Visualization of the task graph for an IT project to design the "Smart House" information system

During Stage 3 of the devised method, 1,000 simulations were conducted to estimate the total duration of the IT project, taking into account delays, using the developed application implementation of the method. For each simulation, the total duration of the project was calculated, taking into account the predicted risks of delays. Visualization of the distribution of the duration of the IT project to design the "Smart House" IS based on the results of Monte Carlo simulations of the devised combined risk assessment method is shown in Fig. 5, *a*.

Then, for control, a simulation of the risks of delays in operations of the same IT project was carried out using the conventional Monte Carlo method. Visualization of the distribution of duration of the IT project to design the "Smart House" IS based on the results of using the conventional Monte Carlo method is shown in Fig. 5, *b*.

A comparison of the results from applying the devised combined method of risk assessment and the conventional Monte Carlo method for assessing the risks of the mentioned IT project is given in Table 2.



a *b*

Fig. 5. Visualization of the distribution of duration of the IT project to design the "Smart House" information system based on the application results: *a* — devised combined risk assessment method; *b* — conventional Monte Carlo method

Table 2

Comparison of applying the devised combined risk assessment method and the conventional Monte Carlo method for risk assessment of the IT project to design the "Smart House" information system

Method	Projected duration (days)	95 % confidence interval (days)
Traditional Monte Carlo method	224.5	206–243
Combined method (SVM+Monte Carlo)	234.5	226–244

Comparison of our results shows that the combined method provides more accurate estimates of project duration and a narrower confidence interval, which indicates higher reliability of forecasts.

6. Discussion of results based on devising an ensemble of ML-methods and the Monte Carlo method for risk assessment of an IT project to design "Smart House" systems

In the process of devising an ensemble of ML methods and the Monte Carlo method, a combined risk assessment method was developed. The essence of this method is to apply ML methods such as SVM and Bayesian networks not to process the results of Monte Carlo simulations but to process the raw data for that simulation. This use of ML methods was due to the main hypothesis put forward during the research, according to which ML methods in the process of risk assessment are better used to improve the initial data in order to eliminate the shortcomings of existing assessment methods. The results of devising the combined method and its experimental verification allow us to assert that this hypothesis has been confirmed.

For this purpose, the features of applied implementation of the devised method were determined. For the software, which ensures the implementation of the devised method, the technological stack was defined, and the fragments of the software code were given. These fragments describe the implementation of the following important elements of the devised method:

– preparation of historical data (Fig. 1);
– SVM model training (Fig. 1);
– performance of model forecasting and evaluation tasks (Fig. 1);
– preparation of project data (Fig. 2);
– forecasting the risks of delays (Fig. 2);
– construction of a graph of project tasks (Fig. 2);
– conducting Monte Carlo simulations (Fig. 3);
– visualization of the simulation results (Fig. 3).

The use in the devised method of the Python programming language and the libraries specified in the technology stack for programmatic implementation made it possible to implement the elements of this method by means that have been repeatedly tested in practice when solving similar problems.

The experimental verification of the devised combined method of risk assessment was carried out on the data of an IT project to design the "Smart House" IS. According to the evaluation results, it was established (Fig. 5, *b*) that the expected duration of this IT project is 234.5 days with a possible range of deviations of 226–244 days (with a 95 % confidence interval).

For comparison, a simulation of the risks of delay in the implementation of the same IT project was carried out using the conventional Monte Carlo method. The simulation results and their comparison with the results of the application of the devised method are shown in Fig. 5 and in Table 2. The results of comparative modeling indicate that the combined method provides more stable and predictable results. In particular, the devised method more accurately identifies tasks that have a high probability of delays, which makes it possible to better allocate resources and focus on critical tasks. This ensures more efficient project management and reduces the risk of delays. The confidence interval for the combined method was significantly narrower compared to the conventional Monte Carlo method. This means that the combined method is able to provide more confident predictions about the duration of the project. A narrower confidence interval indicates greater accuracy and predictability of forecasts, which is critical for effective IT project management.

The application of the devised combined method for risk assessment has another advantage over the results of current studies [12, 13, 16, 17], which are based on the application of ML methods mainly for processing the obtained modeling results. This advantage consists in the absence of time spent on understanding the results of the application of ML methods and translating these results from a numerical description into natural language terms. This method could be used even by those analysts who have only minor competencies in the field of applying ML methods and interpreting their results.

A serious limitation of the application of the devised combined method for risk assessment is the need to build and constantly update a specialized data warehouse, which contains historical data about previous IT projects. But such repositories have become mandatory components of management systems at IT companies in recent decades and are increasingly being used for the automated execution of a new process – the project knowledge management process [10]. Therefore, this limitation should not greatly affect the application of the devised method, especially in IT companies that perform a significant number of IT projects of the same type, large and medium IT projects.

Among other limitations for applying the devised combined risk assessment method, the following should be noted:

– significant costs of computing resources in connection with the use of such a method as SVM;

– the need to spend time on the selection of data sets of previous IT projects analogous to the IT project whose risks are assessed in the data warehouse;

– possible problems with the integration of implementation of the devised method into the existing management systems of IT projects to design the "Smart House" ISs at various IT companies.

It should be noted that the issue of integration of the developed software implementation of the combined method into existing IT project management systems is not considered in this study. This issue requires a separate study of the most common IT project management systems for the development of the Smart House ISs and their external interfaces that provide communication with other software products. The evaluation results obtained using the developed software implementation of the combined method are documented by the analyst and transferred in the form of a paper or electronic document to the manager for further planning of the IT project.

Further area of our research could involve several fields. One is a comparative analysis of the accuracy and costs in applying the proposed combined method and modern assessment methods. Conducting such an analysis would allow us to determine which combination of ML methods and nonML methods is the best in terms of increasing accuracy and reducing costs for solving the evaluation problem. The second direction is conducting research into existing dependences, which characterize risks of various nature. Conducting research in this area makes it possible to replace in the combined method such an expensive ML method as SVM with simpler methods for solving the problem of regression analysis and forecasting. Another direction of research is to study the possibility of applying the devised method for risk assessment of other types of IT projects and projects in other fields of activity. Such studies will make it possible to establish the degree of universality of the devised method and to compile recommendations for its application for risk assessment in other types of projects.

7. Conclusions

1. A combined risk assessment method using ML methods and the Monte Carlo method has been devised. SVM and Bayesian network methods were used as ML methods. The essence of the combined method is to apply SVM and Bayesian networks methods to process raw data about an IT project and its parameters with the aim of further using the obtained processing results as raw data for Monte Carlo simulations. Descriptions of the stages and steps of the combined method are given, the specifici of each of the described steps are considered.

2. We have identified features of applied implementation of the devised method. In particular, the technological stack of the software implementation of the devised method was described. Underlying this stack, it is proposed to use the Python programming language and libraries that extend the capabilities of this language. A software implementation of the method was developed using the elements of this technological stack. Fragments of the program code are given, which describe the implementation of the main elements of the devised method.

3. An experimental verification of the devised method was carried out using an example of the IT project to design a "Smart House" IS. The results of the evaluation determine the expected duration of this IT project at 234.5 days, with a possible deviation range of 226–244 days (at a 95 % confidence interval). For comparative analysis, the delay risk of the same IT project was assessed using the conventional Monte Carlo method. As a result of applying the conventional Monte Carlo method, the expected duration of the IT project was determined to be 224.5 days with a possible deviation range of 206–243 days (at a 95 % confidence interval). The comparison of our data shows that the devised combined method provides higher reliability of forecasts.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Gaied, A. (2020). Smart Home for beginners: Designing and building a smart house. Books on Demand, 81.
2. Evenson, A. (2024). How to Build you Own Smart Home: A simple guide for beginners to upgrade and automate any house system.
3. The standard for risk management in portfolios, programs, and projects (2019). Project Management Institute, 175.
4. Hamilton, S. (2017). How to Make Your Home a Smart Home. Createspace Independent Publishing Platform, 238.

5. Morozova, O., Tetskyi, A., Nicheporuk, A., Kruvak, D., Tkachov, V. (2022). Smart home system security risk assessment. Computer Systems and Information Technologies, 3, 81–88. https://doi.org/10.31891/csit-2021-5-11
6. Hopkin, P., Thompson, C. (2021). Fundamentals of Risk Management: Understanding, Evaluating and Implementing Effective Risk Management. Kogan Page, 480.
7. Jacobsson, A., Boldt, M., Carlsson, B. (2016). A risk analysis of a smart home automation system. Future Generation Computer Systems, 56, 719–733. https://doi.org/10.1016/j.future.2015.09.003
8. Tereshchenko, L. (2021). Risk management of information systems: stages of the risk management process. Economy and Society, 31. https://doi.org/10.32782/2524-0072/2021-31-12
9. Kurdupov, O. L., Potapova, N. A. (2021). Vykorystannia metodu Monte-Karlo dlia analizu proektnykh ryzykiv. Materialy II Vseukrainskoi naukovo-praktychnoi konferentsiyi "Kompiuterni tekhnolohiyi obrobky danykh". Vinnytsia, 138–141. Available at: https://jktod.donnu.edu.ua/article/view/11711
10. ISO/IEC/IEEE 15288:2023. Systems and software engineering. System life cycle processes. https://doi.org/10.1109/ieeestd.2023.10123367
11. Rashid, C. H., Shafi, I., Ahmad, J., Thompson, E. B., Vergara, M. M., de la Torre Diez, I., Ashraf, I. (2023). Software Cost and Effort Estimation: Current Approaches and Future Trends. IEEE Access, 11, 99268–99288. https://doi.org/10.1109/access.2023.3312716
12. Ramaekers, R., Silhavy, R., Silhavy, P. (2023). Software Cost Estimation Using Neural Networks. Software Engineering Research in System Science, 831–847. https://doi.org/10.1007/978-3-031-35311-6_77
13. Kumar, B. N. R., Suresh, Y. (2022). Effective ANN Model based on Neuro-Evolution Mechanism for Realistic Software Estimates in the Early Phase of Software Development. International Journal of Advanced Computer Science and Applications, 13 (2). https://doi.org/10.14569/ijacsa.2022.0130223
14. Nassar, A. H., Elbisy, A. M. (2024). A Machine Learning Approach to Predict Time Delays in Marine Construction Projects. Engineering, Technology & Applied Science Research, 14 (5), 16125–16134. https://doi.org/10.48084/etasr.8173
15. Aljorany, H. O., Mahjoob, A. M. R. (2024). Establishing a Budget for Optimal Response Strategies for Risks Categorized into Distinct Groups by using a Mathematical Model and Genetic Algorithm. Engineering, Technology & Applied Science Research, 14 (3), 14747–14753. https://doi.org/10.48084/etasr.7526
16. Khan, T., Faisal, Mohd. (2023). An efficient Bayesian network model (BNM) for software risk prediction in design phase development. International Journal of Information Technology, 15 (4), 2147–2160. https://doi.org/10.1007/s41870-023-01244-4
17. Bilousiva, L., Gryzun, L., Zhytienova, N., Pikalova, V. (2023). Issues of Formalization of Risk Management Process in Software Design. Proceedings of the 4th International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, 48–57. Available at: https://ceur-ws.org/Vol-3373/keynote4.pdf
18. Bekishev, Y., Pisarenko, Z., Arkadiev, V. (2023). FMEA Model in Risk Analysis for the Implementation of AGV/AMR Robotic Technologies into the Internal Supply System of Enterprises. Risks, 11 (10), 172. https://doi.org/10.3390/risks11100172
19. Albany, M., Alsahafi, E., Alruwili, I., Elkhediri, S. (2022). A review: Secure Internet of thing System for Smart Houses. Procedia Computer Science, 201, 437–444. https://doi.org/10.1016/j.procs.2022.03.057
20. Harkai, A. (2024). Managing cyber-security risks associated with IoT devices for conducting financial transactions within the smart home ecosystem. Procedia Computer Science, 242, 200–210. https://doi.org/10.1016/j.procs.2024.08.260
21. Garlick, A. (2007). Monte Carlo. Estimating Risk, 213–232. https://doi.org/10.4324/9781315256122-25
22. Paltrinieri, N., Comfort, L., Reniers, G. (2019). Learning about risk: Machine learning for risk assessment. Safety Science, 118, 475–486. https://doi.org/10.1016/j.ssci.2019.06.001
23. Hovorushchenko, T., Aleksov, S., Talapchuk, S., Shpylyuk, O., Magdin, V. (2022). Overview of the methods and tools for situation identification and decision-making support in the cyberphysical system "smart house". Computer Systems and Information Technologies, 4, 20–26. https://doi.org/10.31891/csit-2022-4-3
24. Abdallah, Z. S., Du, L., Webb, G. I. (2023). Data Preparation. Encyclopedia of Machine Learning and Data Science, 1–10. https://doi.org/10.1007/978-1-4899-7502-7_62-2
25. Li, B., Lu, P. (2024). Normalize Data component. Azure Machine Learning. Available at: https://learn.microsoft.com/uk-ua/azure/machine-learning/component-reference/normalize-data?view=azureml-api-2
26. Jo, T. (2020). Data Encoding. Machine Learning Foundations, 47–68. https://doi.org/10.1007/978-3-030-65900-4_3
27. Kelleher, J. D. (2019). Deep Learning. The MIT Press. https://doi.org/10.7551/mitpress/11171.001.0001
28. Leshchuk, H. V. (2017). Simulation Analysis of Monte Carlo in the System of Investment Projects Risks Assessment. Ukrainian Journal of Applied Economics, 2 (1), 57–67. Available at: http://ujae.wunu.edu.ua/index.php/ujae/article/view/119
29. Novkov, A. (2024). Monte Carlo Charts: How to Use Them in Practice? Available at: https://businessmap.io/blog/monte-carlo-chart-graph
30. Yak vybraty pravylnyi tekhnolohichnyi stek dlia vashoho proektu. REDSTONE. Available at: https://redstone.agency/blog/yak-vybraty-pravylnyi-tekhnolohichnyi-stek-dlia-vashoho-proektu/