https://doi.org/10.31891/csit-2024-2-9
UDC 004.94

Andrii KOPP, Uliya LITVINOVA, Roman LUCHNOI
National Technical University "Kharkiv Polytechnic Institute"

RESEARCH ON SOFTWARE FOR ERROR PROBABILITY PREDICTION IN BUSINESS PROCESS MODELS USING LOGISTIC REGRESSION

Business process modeling allows to graphically represent organizational activities and related events. It allows to identify areas for improving the organizational performance, define requirements for software solutions, and, in general, to facilitate communication between IT and business parties within or between different organizations. Therefore, at the stage of representing the activity in the form of a model, it is necessary to understand how likely it is that errors will occur during the implementation of the depicted business process. Thus, this study aims to improve the quality of business process models by solving the problem of predicting the error probability of business process execution. In order to assign error probabilities to each business process model from the training dataset, it is proposed to use one of the complexity metrics – the coefficient of network connectivity. To predict the error probability in business process execution, it is proposed to use the simplest and most intuitive machine learning model – logistic regression. As independent variables, it is proposed to choose the basic metrics of business process modeling – the number of nodes and arcs. Thus, the algorithm for solving the task includes steps related to calculating probabilities for the training data set, preparing training and test sets, determining regression parameters, visualization, and evaluation of training results. For the software that implements the proposed approach, a client-server architecture was chosen due to its flexibility and scalability. When developing software components, the Scikit-Learn machine learning library and the Python programming language were used to build a logistic regression mathematical model. The software tool is implemented as a web application based on MySQL, the Node JS platform, and the Express JS web framework. The quality assessment results of the developed prediction model indicate the suitability of the software tool for solving the problem of predicting the error probability of business process execution.

Keywords: software tool, error probability prediction, business process model, logistic regression model.

Андрій КОПП, Юлія ЛІТВІНОВА, Роман ЛУЧНОЙ
Національний технічний університет «Харківський політехнічний інститут»

ДОСЛІДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ЙМОВІРНОСТІ ПОМИЛОК У МОДЕЛЯХ БІЗНЕС-ПРОЦЕСІВ З ВИКОРИСТАННЯМ ЛОГІСТИЧНОЇ РЕГРЕСІЇ

Моделювання бізнес-процесів дозволяє графічно зображувати організаційні активності та пов'язані з ними події. Це, в свою чергу, дозволяє визначати напрямки покращення діяльності підприємства, визначати вимоги до програмних рішень та в цілому – для порозуміння ІТ- та бізнес-сторін всередині або поміж різними організаціями. Таким чином, вже на етапі представлення діяльності у вигляді моделі необхідно розуміти наскільки ймовірним є виникнення помилок під час реалізації зображеного бізнес-процесу. Отже, дана робота має на меті підвищення якості моделей бізнес-процесів за рахунок розв'язання задачі прогнозування ймовірності виникнення помилок виконання бізнес-процесів. Для того, щоб для кожної моделі бізнес-процесу з навчального набору даних призначити ймовірності виникнення помилок, пропонується використовувати одну з метрик складності – коефіцієнт структурної зв'язності. Для прогнозування ймовірності виникнення помилок при виконанні бізнес-процесів пропонується використовувати найпростішу та інтуїтивно зрозумілу модель машинного навчання – логістичну регресію. В ролі незалежних змінних пропонується обрати базові метрики моделювання бізнес-процесів – кількість вузлів та дуг. Таким чином, алгоритм розв'язання поставленої задачі включає кроки, пов'язані з обчисленням ймовірностей для навчального набору даних, підготовку навчального та тестового наборів, визначення параметрів регресії, візуалізацію та оцінювання результатів навчання. Для програмного забезпечення, яке реалізує запропонований підхід, було обрано клієнт-серверну архітектуру завдяки її гнучкості та здатності до масштабування. Під час розробки програмних компонентів було використано бібліотеку машинного навчання Scikit-Learn та мову програмування Python для побудови математичної моделі логістичної регресії. Прикладна частина реалізована як веб-застосунок на основі MySQL, платформи Node JS та веб-фреймворку Express JS. Результати оцінювання якості розробленої моделі прогнозування кажуть про придатність програмного забезпечення до розв'язання поставленої задачі прогнозування ймовірності виникнення помилок виконання бізнес-процесів.

Ключові слова: програмний засіб, прогнозування ймовірності помилок, модель бізнес-процесу, модель логістичної регресії.

Introduction

Today, enterprises and organizations aim to ensure the efficiency of their activities in order to increase competitiveness in a dynamic external environment. One of the approaches to organizational management is the business process management approach, which involves considering an enterprise as a connected set of business processes.

A business process is an ordered set of actions (tasks, works, etc.) aimed at transforming incoming material or information resources into services or goods of value to end users. Effective execution of business processes allows organizations to achieve their strategic goals. Thus, the task of business process management becomes even more relevant. One of the most important tools used in the business process management approach is business process modeling. Typically, business process modeling is aimed at creating graphical models that reflect the specifics of business processes, taking into account certain events, tasks, decision points, etc. Business process models are used to document activities to understand by participants, to analyze them by stakeholders in order to find "weaknesses" and improve organizational performance, and to help business and technical specialists understand each other when identifying requirements for information system software designed to automate these processes.

Therefore, already at the stage of building or analyzing a business process model, it is necessary to have an idea of the error probability during its execution. Quality analysis of business process models was covered by many research studies over the last decade. From various quality frameworks of process modeling [1] to quantitative approaches based on structural metrics (mostly originated from graph theory or software code metrics) [2]. In [3] authors prove the negative impact of poorly designed business process models to their execution. Various thresholds to assess the efficiency level of business process models were proposed in [4] and [5]. More structural metrics for business process models that reflect different quality attributes (i.e., understandability, maintainability, complexity etc.) are given in [6].

The study aims to improve the quality of business process models by solving the problem of predicting the error probability of business process execution. The research object is the process of predicting the error probability of business process execution. The research subject is the algorithm and software for predicting the error probability of business execution.

Existing business process modeling and analysis tools

Today, the de facto standard for business process modeling is the Business Process Model and Notation (BPMN).

BPMN is a method of creating diagrams that model the stages of a planned business process from the initial event that starts the process to the final event that ends the process. The BPMN notation is the basis of Business Process Management (BPM) – it visually depicts the detailed sequence of events and tasks required to complete a process [7].

The BPMN notation is used to build easy-to-read graphical models of business processes that can be shared across organizations and industries. BPMN diagram symbols are divided into four main groups [7]:

1) flow objects – events, tasks, sub-processes, gateways;
2) connecting objects – sequence flows, message flows, associations;
3) pools and lanes;
4) artifacts – annotations, groups, data objects.

The core elements of the BPMN business process modeling notation are shown in Fig. 1.

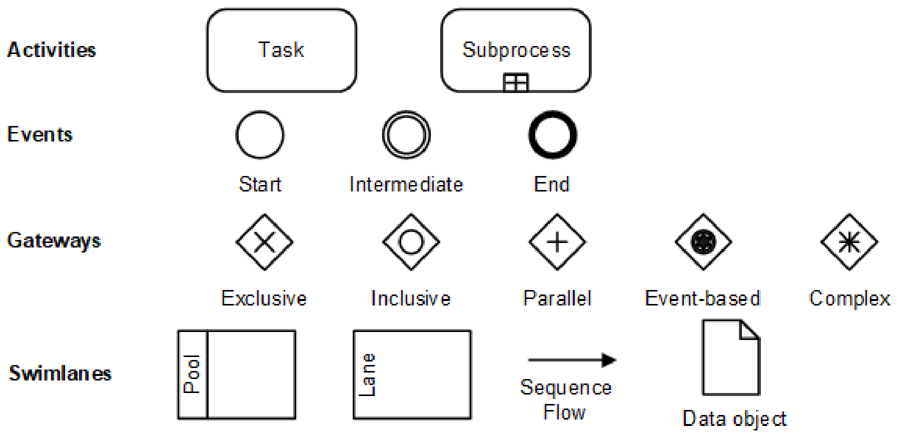

Fig. 1. Core elements of the BPMN business process modeling notation

Therefore, it is reasonable to analyze the tools for business process modeling and analysis in terms of BPMN notation support and quality assurance of the created business process models, in particular, in terms of predicting the error probability of business process execution. A comparative analysis of existing business process modeling and analysis tools is shown in Table 1.

Thus, SAP Signavio Process Manager provides the most capabilities among the most common business process modeling and analysis tools regarding error prediction and notification of users at the stage of BPMN modeling. This tool is based on the business process modeling rules and notifies model authors about mistakes they may make.

At the same time, the presence or absence of errors in the model are discrete values that indicate the quality of the BPMN diagram "here and now", but do not allow to assess the consequences of errors occurring during the execution of the business process. Thus, the problem of predicting the error probability of business process execution is relevant, and the appropriate algorithmic and software should be developed to solve it.

Table 1

Comparative analysis of existing business process modeling and analysis tools

Business process modeling and analysis tool	BPMN modeling notation support	Quality control of created business process models	Simulation of business processes	Business process execution analysis
Appian [8]	Supports	Check for BPMN syntax compliance	Does not support	Supports
Kissflow [9]	Supports	Check for BPMN syntax compliance	Does not support	Supports
Bizagi [10]	Supports	Check the business process structure	Supports	Supports
Signavio [11]	Supports	Comprehensive quality assessment of business process models	Supports	Supports

Problem statement

This paper solves the relevant problem of predicting the error probability of business process execution, for which it is necessary to develop the appropriate algorithm and software.

Thus, to achieve the aim of the study, it is necessary to solve the following tasks:

1) choose the method for predicting the error probability of business process models;
2) develop the algorithm for predicting the error probability of business process execution;
3) analyze the workflow of predicting the error probability of business process execution;
4) design and develop the software for predicting the error probability of business process execution;
5) verify the performance of the developed algorithm and software, analyze and discuss obtained results.

Error probability evaluation of business process models

The complexity of a BPMN model should be limited, both in terms of physical size and in terms of the number of elements and connections contained in the model. Large diagrams tend to indicate an unclear horizontal and vertical separation of the business process [12]. Vertical separation means that the model should fit into one clearly defined level of abstraction. This means that all actions contained in the model have the same level of detail and abstraction. A useful metric for establishing a consistent level of abstraction is the use of process objects, i.e., key data elements, documents, or files that flow through the process and are manipulated during specific activities. If one activity works with a set of documents while the next activity works with an attribute of a data item, these activities are not at the same level of abstraction [12]. Horizontal separation means that an end-to-end process has to be broken down into several connected diagrams if it is not described at the highest level of abstraction (usually as a value chain) [12].

Therefore, too complex business process models reduce their understanding and, as a result, lead to errors in business process execution. The following are levels of probability that a business process model is efficient [5]:

1) level 1 – 10% probability that the model is efficient;
2) level 2 – 30% probability that the model is efficient;
3) level 3 – 50% probability that the model is efficient;
4) level 4 – 70% probability that the model is efficient.

As for the connectivity of the business process model, study [5] defines the following thresholds:

1) level 1 (“very inefficient”) – 1.7;
2) level 2 (“rather inefficient”) – 1.1;
3) level 3 (“rather efficient”) – 0.6;
4) level 4 (“very efficient”) – 0.4.

Therefore, the proposed probability levels and thresholds can be interpreted, for example, as follows: if the connectivity of the elements of the business process model is less than or equal to 0.4, this model is considered “very effective” in terms of its understanding by stakeholders, and the probability of errors during the business process execution is about 10% [5].

Thus, the error probability of business process execution based on the coefficient of network connectivity (CNC) can be determined using the following formulas:

$$CNC = \frac{Arcs}{Nodes}, \tag{1}$$

$$P_{Error} = \begin{cases} 0.1, CNC \leq 0.4, \\ 0.3, CNC \leq 0.6, \\ 0.5, CNC \leq 1.1, \\ 0.7, CNC \leq 1.7, \\ 1.0, CNC > 1.7, \end{cases} \tag{2}$$

where:

– $Nodes$ is the number of business process elements;

– $Arcs$ is the number of business process sequence flows.

To calculate the number of elements and sequence flows in a business process model represented in the BPMN notation, it is proposed to process the corresponding model file. According to the specification [13], business process model files are represented as XML (eXtensible Modeling Language) files according to the document schema defined by the BPMN standard. Therefore, based on the BPMN standard and the types of objects that can be included in the business process description, we need to get the number of “FlowNode” elements from the XML file containing the business process definition to find the value $Nodes$ [13]:

1) activities (“Activity”);
2) events (“Event”);
3) gateways (“Gateway”).

To find the value $Arcs$, we need to get the number of “SequenceFlow” elements – sequence flows, which in a BPMN file specify connections between business process elements using “sourceRef” and “targetRef” properties [13].

Algorithm for error probability prediction in business process models

Logistic regression is one of the most popular machine learning algorithms that belongs to the supervised learning methods. It is used to predict a categorical dependent variable using a given set of independent variables. Logistic regression produces probability values between 0 and 1 [14]. In this study, logistic regression is proposed to be used to predict the error probability of business process execution using the essential BPMN model structural features. The logistic regression method establishes a relationship between the dependent (y) and one or multiple independent (x) variables, but instead of a regression line, an S-shaped logistic function is used to predict two maximum values (0 or 1). Thus, logistic regression shows the probability of something – it determines how the value of the dependent variable y changes in response to the value of the independent variable x [14].

Mathematically, the logistic regression model for the problem of error probability prediction in business process models can be described by the following formula [14]:

$$P_{Error} = \frac{1}{1+e^{-(a_0+a_1 Nodes+a_2 Arcs)}}, \tag{3}$$

where a_0, a_1, and a_2 are the logistic regression parameters that should be found.

To determine the parameters of logistic regression, the minimization of the residual sum of squares MSE (Mean Squared Error) is used [14]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left[P_{Error}^{i} - \frac{1}{1+e^{-(a_0+a_1 Nodes_i+a_2 Arcs_i)}}\right]^2 \to \min_{a_0,a_1,a_2}, \tag{4}$$

where:

– n is the number of observations;
– $P_{Error}^{i}(CNC)$ is the error probability in each observation, $i=\overline{1,n}$;
– $Nodes_i$ is the number of business process elements in each observation, $i=\overline{1,n}$;
– $Arcs_i$ is the number of business process sequence flows in each observation, $i=\overline{1,n}$.

To minimize MSE, the gradient descent method [15] is usually used:

$$a_j = a_j - \alpha\frac{\partial}{\partial a_j}J(a_0,a_1,a_2), j=0,1,2, \tag{5}$$

where $J(a_0,a_1,a_2)$ is the cost function, basically MSE (4).

However, in this paper, we propose to use machine learning software libraries where the logistic regression method is already implemented [14].

The proposed algorithm (Fig. 2) for solving the problem of predicting the error probability of business process execution includes the following steps:

1) calculate the probabilities using expression (2) for a set of business process models;
2) prepare training and test datasets that include independent variables (number of nodes and arcs) and estimate the error probability of a business process execution;
3) determine the parameters of logistic regression (3) by minimizing the MSE (4);
4) evaluate the logistic regression model for the training and test data sets;
5) determine the business process model features for which the error probability should be predicted;
6) predict the error probability using the trained logistic regression model (3).

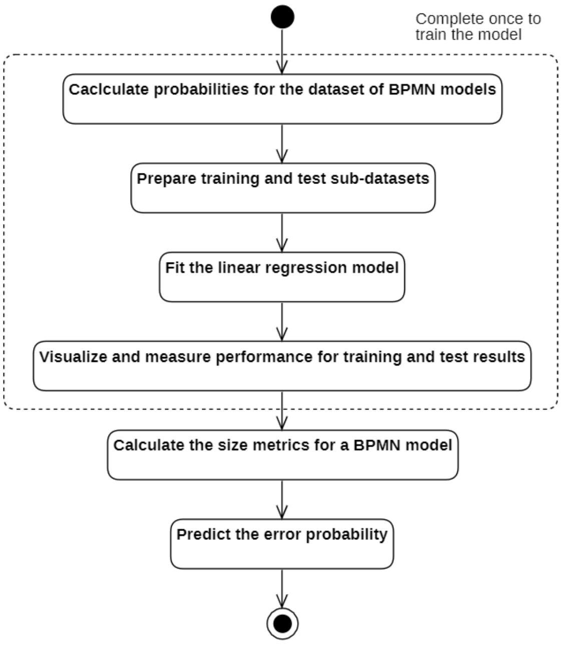

Fig. 2. Algorithm for predicting the error probability of business process execution

Workflow of error probability prediction in business process models

The workflow (Fig. 3) of predicting the error probability of business process execution is analyzed using the graphical notation for functional modeling IDEF0. The prediction workflow involves a data analyst and a business analyst who use the software tool to solve the problem. The inputs to the workflow are a dataset of business process models collected as part of the BPMAI project [16] and a BPMN model to be analyzed. As a result of the prediction workflow, an estimate of the error probability of the business process execution and a corresponding report are provided. The prediction workflow is based on the BPMN meta-model [13], machine learning models and probability thresholds [5].

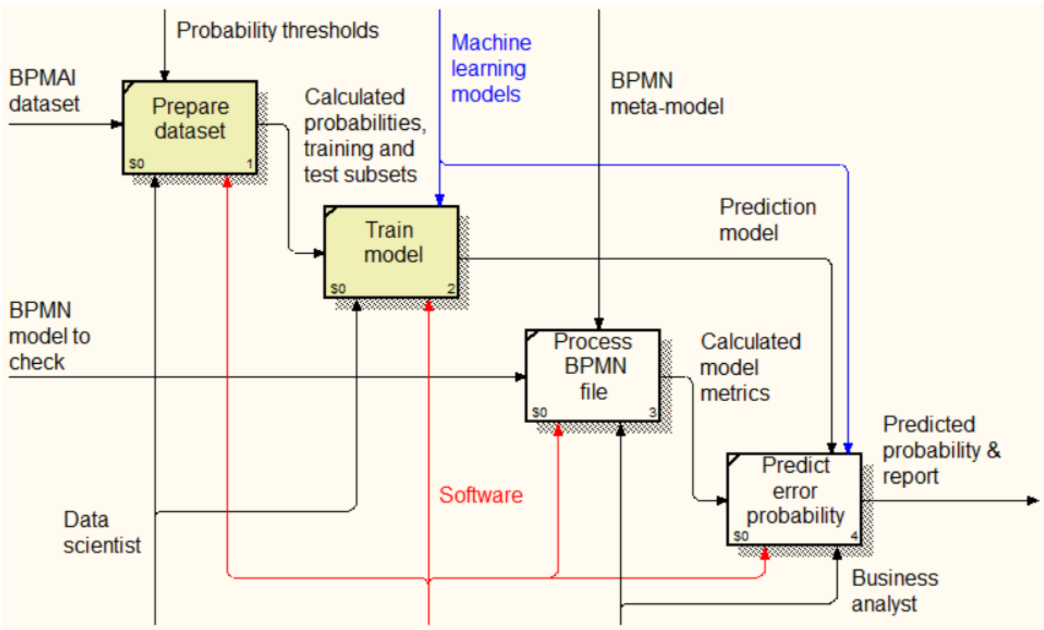

Fig. 3. Workflow diagram of the error probability prediction in business process models

A detailed description of the workflow for predicting the error probability of business process execution includes activities related to preparing the data set, training the model, processing the BPMN file uploaded by the user, and making predictions and preparing the corresponding report (Fig. 3). Steps 1 – 2 of this workflow, highlighted in yellow in Fig. 3, as well as steps 1 – 4 of the algorithm (Fig. 2) are performed only at the training stage of the machine learning model.

Software tool for error probability prediction in business process models

The client-server architectural model [17] was chosen to develop the software for predicting the error probability of business process execution due to its advantages of scalability, security, and availability. Therefore, Fig. 4 demonstrates a software components deployment diagram created according to the selected type of a system architecture.

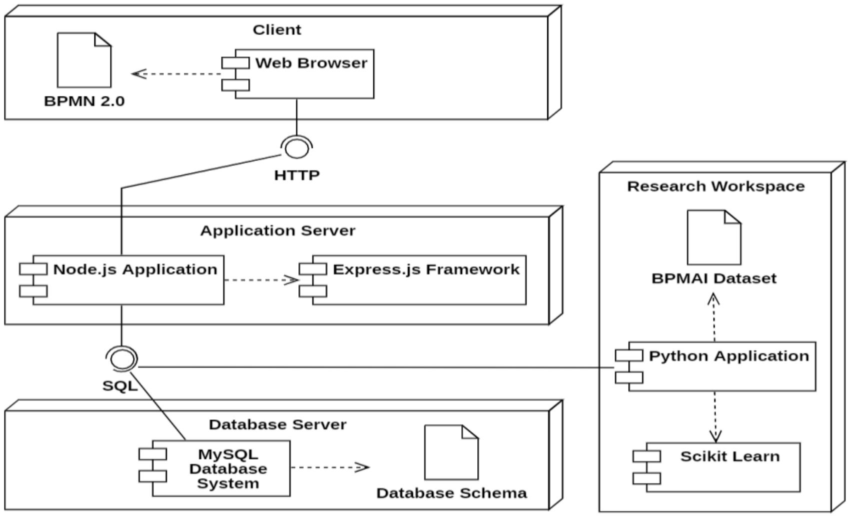

Fig. 3. Software components deployment diagram

To build a classification model based on logistic regression (Fig. 3) and predict the error probability of business process execution, it is proposed to use the Python programming language and the Scikit-Learn library [18]. As for the application for predicting the likelihood of business process errors, it is proposed to use the Node JS [19] platform (to create a server-side web application in JavaScript) and the Express JS [19] framework (to simplify the development of a web application), as well as the MySQL database management system (DBMS) the second popular DBMS [20].

The proposed use case diagram depicts the capabilities of the software to meet the needs of its users - data analysts and business analysts:

1) using the data (from the BPMAI collection [16]) prepared for loading and processing;

2) calculating the error probabilities of business process execution and building a prediction model, including partitioning the dataset for training and testing, as well as feature scaling;

3) uploading BPMN models, predicting the error probability of business process execution, and formulating recommendations for improving business process models.

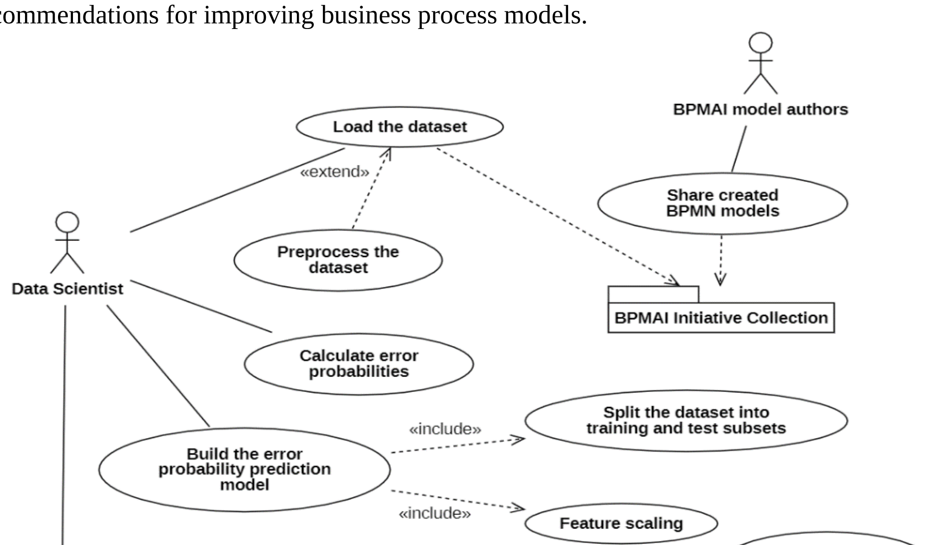

Fig. 4. Software components deployment diagram

Software users have access to a ribbon with previously uploaded BPMN models (Fig. 5).

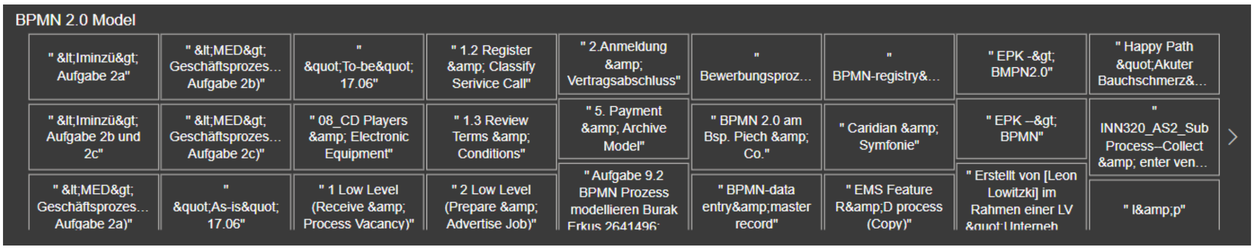

Fig. 5. Selection among uploaded business process models for analysis

For the selected model, its name is displayed and a pie chart with the number of core business process elements is provided (Fig. 6).

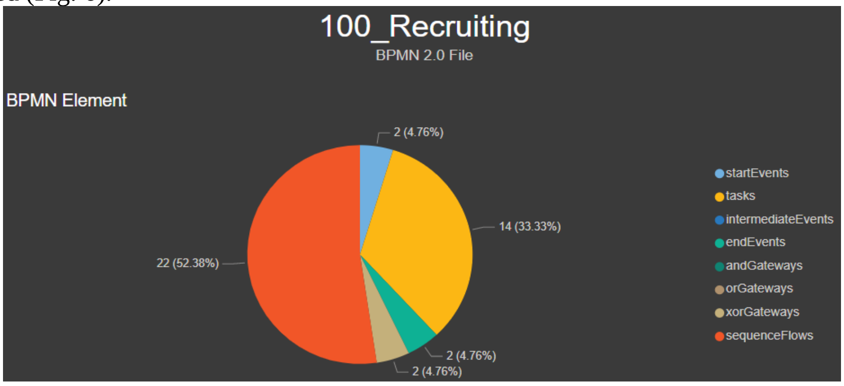

Fig. 6. Structure of the selected BPMN model

Also, for the selected business process model, the value of the coefficient of network connectivity (1) and the error probability of a business process are given (Fig. 7).

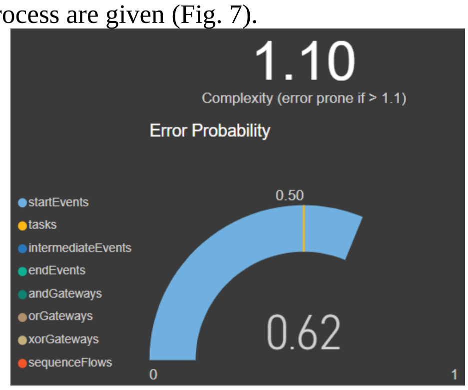

Fig. 7. CNC and predicted error probability of business process execution

Results and discussion

To verify the performance of the developed software in predicting the error probability of business process execution, we used the business process models from the BPMAI [8]. Based on CNC (1), models from this dataset were previously analyzed for possible presence of errors. A dataset from the BPMAI project [16] containing descriptions of business process models has already been prepared and is available as a CSV (Comma-Separated Values) file.

The BPMAI project [16] defines descriptions of 18812 business process models of different organizations from different countries (Fig. 8):

– training dataset for building a logistic regression model is 80% of the total dataset – 15049 models;

– test dataset is 20% of the total data set – 3763 models.

Using the Scikit-Learn library of the Python programming language and the training dataset, a logistic regression model was built to predict the error probability of business process execution:

$$P_{Error} = \frac{1}{1+e^{0.54-6.09\cdot Nodes+2.65\cdot Arcs}}. \quad (6)$$

Hence, the following logistic regression parameters minimize the residual sum of squares (MSE) given in (4):

1) $a_0 = -0.54$;
2) $a_1 = 6.09$;
3) $a_2 = -2.65$.

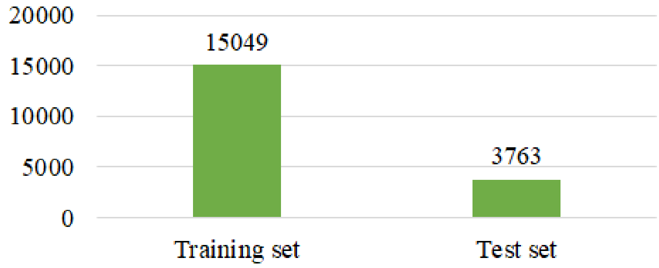

Fig. 8. The distribution of BPMN model data by training and test sets

To find the regression parameters, one of the multivariate optimization methods could be used, for example, the gradient descent method (5). However, in this work, we used the Scikit-Learn machine learning package, which already implements a classification method based on logistic regression [18].

The classification results are shown in Fig. 9. The blue color (category "1") represents BPMN models that have been identified as error-prone models. To analyze the performance of the developed algorithm and software for detecting structural errors in business process models, we will use the confusion matrix (Table 1) [21].

Table 1

The confusion matrix for logistic regression performance analysis

Does a business process model have structural errors?		**Correct answers**	
		Yes	**No**
Classification results	Yes	2281	211
	No	420	851

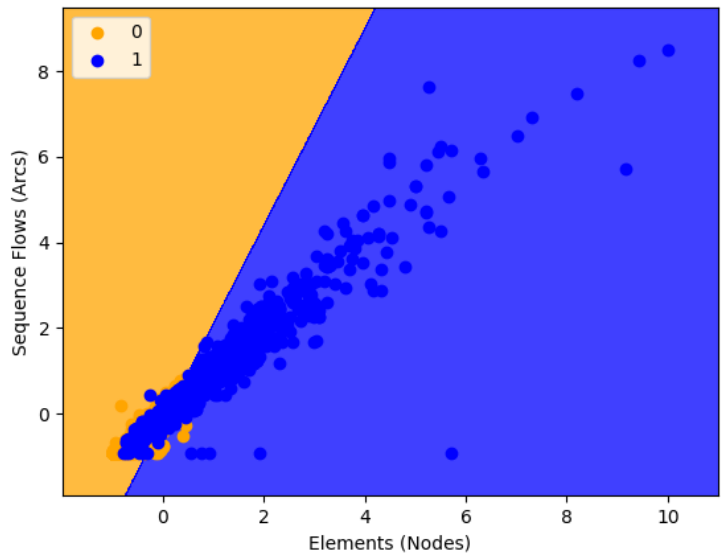

Fig. 9. The results of BPMN models classification using logistic regression

Based on the obtained confusion matrix, the following indicators were determined [21]:
1) true-positive (TP) results – 2281 models;
2) false-positive (FP) results – 211 models;
3) false negative (FN) results – 420 models;
4) true-negative (TN) results – 581 models.
Therefore, the following quality metrics [21] of the created machine learning model were computed:
1) precision (the share of correct answers within a class):

$$Precision = \frac{TP}{TP+FP} = 0.92; \tag{7}$$

2) recall (the share of true-positive classifications):

$$Recall = \frac{TP}{TP+FN} = 0.84; \tag{8}$$

3) accuracy (share of correct answers):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = 0.83; \tag{9}$$

4) summarized performance metric:

$$F - measure = 2\frac{Precision \times Recall}{Precision + Recall} = 0.88; \tag{10}$$

The obtained values of precision (7) and recall (8), as well as accuracy (9) and F-measure (10) are quite high, which makes it possible to consider the developed algorithm and software as suitable for predicting the error probability of business process execution.

Conclusions

This study addresses the relevant practical problem of predicting the error probability of business process execution. Thus, the following tasks were solved in this paper:

1) logistic regression is chosen for predicting the probability of business process errors;
2) developed the algorithm for predicting the error probability of business process execution;
3) analyzed the workflow of predicting the error probability of business process execution;
4) developed the software tool for predicting the error probability of business process execution;
5) verified the performance of the developed software and analyzed the obtained results.

Obtained results allow us to conclude that it is possible to predict the presence of errors in the execution of business processes by analyzing the model complexity – complex models are incomprehensible and inefficient to use and can lead to error occurrence in the processes themselves. Thus, the developed tool can be used to analyze the complexity of BPMN 2.0 models and, accordingly, the error propensity.

References

1. Pavlicek J. The business process model quality metrics / J. Pavlicek, R. Hronza, K. Jelinkova // Enterprise and Organizational Modeling and Simulation, EOMAS 2017. – Lecture Notes in Business Information Processing : Springer, Cham. – 2017. – Vol. 298. – P. 134-148.
2. Corradini F. Quality assessment strategy: Applying business process modelling understandability guidelines / F. Corradini, F. Fornari, S. Gnesi, A. Polini, B. Re // University of Camerino : Italy. – 2015. – 98 p.
3. Henrik L. Learning from quality issues of BPMN models from industry / L. Henrik, J. Mendling, O. Günther // IEEE Software. – 2015. – Vol. 4, N. 33. – P. 26-33.
4. Kbaier W. Determining the threshold values of quality metrics in BPMN process models using data mining techniques / W. Kbaier, S. A. Ghannouchi // Procedia Computer Science. – 2019. – Vol. 164. – P. 113-119.
5. Sánchez-González L. Quality assessment of business process models based on thresholds / L. Sánchez-González, F. Garcia, J. Mendling, F. Ruiz // On the Move to Meaningful Internet Systems : Springer, Berlin. – 2010. – P. 78-95.
6. Ayech H. B. H. Extension of the BPM lifecycle to promote the maintainability of BPMN models / H. B. H. Ayech, S. A. Ghannouchi, E. A. E. H. Amor // Procedia Computer Science. – 2021. – Vol. 181. – P. 852-860.
7. What is Business Process Modeling Notation?, https://www.lucidchart.com/pages/bpmn, 9.11.2023.
8. Platform Overview, https://appian.com/platform/overview.html, 9.11.2023.
9. 5 Top Business Process Analysis Tools Reviewed, https://kissflow.com/workflow/bpm/top-business-process-analysis-tools/, 9.11.2023.
10. Bizagi BPM Suite, https://www.bpmleader.com/bizagi-bpm-suite/, 9.11.2023.
11. SAP Signavio Process Manager – BPM Platform for Process Modeling, https://www.signavio.com/products/process-manager/, 9.11.2023.
12. Enterprise Architecture based on Design Primitives and Patterns, https://dodcio.defense.gov/portals/0/documents/dodaf/primitives_ov-6c_guidelines.pdf, 18.11.2023.
13. BPMN Specification – Business Process Model and Notation, https://www.bpmn.org/, 18.11.2023.
14. Molnar C. Interpretable Machine Learning / C. Molnar // Leanpub. – 2020. – 320 p.
15. Watt J. Machine Learning Refined: Foundations, Algorithms, and Applications / J. Watt, R. Borhani, A. K. Katsaggelos // Cambridge University Press. – 2020. – 573 p.
16. Model Collection of the Business Process Management Academic Initiative, https://doi.org/10.5281/zenodo.3758704, 12.11.2023.
17. Goodyear M. / Enterprise System Architectures: Building Client Server and Web Based Systems / M. Goodyear // CRC Press. – 2017. – 978 p.
18. Hackeling G. Mastering Machine Learning with Scikit-learn / G. Hackeling // Packt Publishing. – 2017. – 254 p.
19. Mardan A. Practical Node.js: Building Real-World Scalable Web Apps / A. Mardan // Apress. – 2018. – 505 p.
20. DB-Engines Ranking, h https://db-engines.com/en/ranking, 24.03.2024.
21. Brownlee J. Imbalanced Classification with Python. Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning / J. Brownlee // Machine Learning Mastery. – 2020. – 463 p.

Andrii KOPP **Андрій КОПП**	PhD, Associate Professor, Head of Software Engineering and Management Intelligent Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: kopp93@gmail.com. https://orcid.org/0000-0002-3189-5623, Scopus Author ID: 57202887287, Researcher ID: Q-7501-2019	доктор філософії, доцент, завідувач кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.
Uliya LITVINOVA **Юлія ЛІТВІНОВА**	PhD in Engineering, Associate Professor of Software Engineering and Management Intelligent Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: litjuli56@gmail.com. https://orcid.org/0000-0001-6680-662X, Scopus Author ID: 58759286800	кандидат технічних наук, доцент кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.
Roman LUCHNOI **Роман ЛУЧНОЙ**	Student of Software Engineering and Management Intelligent Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: roman.luchnoi@cs.khpi.edu.ua	студент кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна