MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 4, pp. 1025–1034 (2024)

Application of the Bayesian approach to modeling credit risks

Senyk A. P.[1], Manziy O. S.[1], Ohloblin P. E.[1], Senyk Y. A.[2], Krasiuk O. P.[3]

[1]*Lviv Polytechnic National University, 12 S. Bandera Str., 79013, Lviv, Ukraine*
[2]*Lviv Forestry University of Ukraine, 103 Chuprinka Str., 79057, Lviv, Ukraine*
[3]*Hetman Petro Sahaidachnyi National Army Academy, 32 Heroes of Maidan Str., 79026, Lviv, Ukraine*

(Received 20 March 2024; Revised 15 November 2024; Accepted 17 November 2024)

A computer model for analyzing, evaluating, and forecasting bank credit risks has been developed. Utilizing a Bayesian network (BN) and established parameter estimation methods, this model was implemented in the Python programming language. It predicts the probability that a borrower may fail to meet financial obligations, such as repaying a loan.

Keywords: *mathematical methods; data analysis; graph; probabilistic parameters; machine learning; credit risk models; Bayesian network; Bayesian approach.*

2010 MSC: 91B30 **DOI:** 10.23939/mmc2024.04.1025

1. Introduction

In the field of modern finance, credit risk modeling and quantification is a necessary and important task. Credit risk assessment consists in calculating the probability that the borrower will not fulfill his financial obligations, such as, for example, repayment of the loan. For financial institutions, this is a key process, as it serves as the basis for making informed decisions. In addition, credit risk assessment plays an important role in maintaining the stability and integrity of financial markets.

The definition of default (DoD) requirements relates to how banks recognize credit defaults and include quantitative analysis and time-to-default, which affect the validity of risk models and processes. The idea presented in [1] is to use a Bayesian approach in different credit risk modeling contexts.

Credit risk assessment is inherently a complex and complex task. This requires synthesizing a large amount of diverse information, such as financial and economic indicators and other factors, into a coherent structure. The BNs appear to be a powerful analytical tool to solve such a task. A BN is a probabilistic graphical model that provides a formal, holistic representation of knowledge and information that contains uncertainty. This mathematical model can reflect the interdependencies and conditional probabilities that underlie credit risk. Using a BN, it is possible to model the relationships between various factors affecting credit risk, such as income level, employment, credit history, etc. The graphical representation of BNs allows transparent visualization of these complex relationships, aiding in decision-making and risk management.

A review of modern sources confirms the effectiveness of applying the Bayesian approach to solving risk management problems. Thus, in the article [2], a discrete BN with a latent variable and a built-in clustering function was used to model non-payments of credit subscribers. Paper [3] presents a Bayesian posterior density (PD) estimation approach and compares it with the classical approach. Paper [4] proposes a BN model to solve credit risk assessment. A new approach to credit risk assessment proposed in [5] is a dynamic BN-based decision support method that can improve decision-makers ability to predict risks in advance. Paper [6] suggests using algorithms to model highly unbalanced credit fraud data. Bayesian optimization is used to find the parameters and determine the optimization function for the model, which is tested on real European credit card fraud data. Paper [7] uses Bayesian hypothesis testing to illustrate the relationship between the limited accuracy of a credit rating agency and its disclosure of an issuer's credit risk at specific time intervals. Paper [8] explores the key advantages of the Bayesian approach and how it can be used to predict loan defaults in the banking sector. Based on the popular logistic credit scoring model, the paper [9] discusses in detail the process of building a Bayesian Model Average (BMA) credit risk quantification model, and then this paper uses three performance measures to compare the forecasting performance between BMA and other methods.

This short review confirms that Bayesian methods have emerged as an alternative to classical approaches. In these and many other sources, the BN model is shown to perform well against competing models (logistic regression model and neural network model) on several parameters such as accuracy, sensitivity, and precision. In the proposed article, an attempt is made to demonstrate that Bayesian methods are really powerful in financial data analytics and applications.

An applied result of many studies in this area is the development and creation of information systems to support decision-making in the analysis of financial and investment risks. A number of works by Oleksandra Manzii and Andriy Senyk [10–12] are devoted to the description of such systems that allow potential investors to independently assess the effectiveness of the investment portfolio set by comparing the growth dynamics of stocks available on the financial market. The proposed information products use a visualization process that presents available tabulated information in a structured form of schemes, graphs, charts, and also use modern mathematical methods and artificial intelligence tools to forecast the riskiness and profitability of the selected investment portfolio.

Graphs are a powerful tool for researching risk management problems and not only that. The perspective of the application of graph theory, which is focused on the display of graph properties and risk components, is considered in [13]. In this paper, the hazard intensity variable is determined by the share of affected nodes among all nodes forming the network. The tendency of a node to be isolated is represented by its proximity, betweenness and degree of network distribution. The article [14] proposes a controlled model of the risk assessment system based on graph theory. Using the bipartite graph technique, the risk assessment due to the propagation of the attack is assumed. The proposed structure is verified by empirical analysis and experiments. An example of a graphical network applied to risk assessment can be seen in the World Economic Forum Report on The Global Risks Interconnections 2020, where the traditional approach was taken a step further and survey respondents were asked to select up to six pairs of global risks that they considered most interconnected.

In the section below, we will try to provide a basic example of how you can use graph network algorithms in risk assessment to analyze and classify risks using the Python programming language. A comparative analysis of the characteristics and capabilities of Python, Java and C# programming languages configured for data analysis in order to choose the optimal one is proposed in [15].

Inefficient risk management can lead to significant losses for financial institutions, therefore, in order to improve the effectiveness of risk management, modern tools and technologies are used. Research on this topic is presented in the works of scientists from Canada (Zhe Gao and Qigang Gao from the Faculty of Computer Science, Dalhousie University, Canada), Korea (Kyungwon Kim, Eun Kwon and Jaram Park from AI Center of Samsung Research, Samsung Electronics Company, Ltd., Republic of Korea) and other outstanding technical scientists.

The development of new mathematical and software algorithms for the analysis and forecasting of banking risks is an urgent task.

2. Statement of the problem

Software solutions that provide an opportunity to automate the evaluation process allow financial institutions to increase the accuracy of the process of analyzing existing risks and the efficiency of decision-making regarding their elimination. The characteristic features of such tools are:

— Using artificial intelligence to automate processes and support decision-making;
— Ability to access various data sources for further analysis;
— Calculation of a wide range of credit risk indicators.

In this work, the authors propose the use of a BN to assess the risks of non-return of payments to a credit institution.

The BN is a structure in the form of a directed acyclic graph, which requires defining a set of vertices and establishing connections between them. Each vertex of the graph corresponds to a random variable, which is limited by the number of possible states, and these states are mutually exclusive. Tables of conditional probabilities of occurrence of certain states of the variable are also defined for each variable network. Tables of conditional probabilities, as well as the structure of the BN, are set manually using expert knowledge or are calculated using machine learning algorithms based on a specific set of data. The processes of building a structure (or studying it based on data) and determining parameters (or studying them based on data) collectively form a complex process of developing and training a BN.

2.1. Initial processing of input data

The dataset for training the BN is taken from Kaggle, an open-source platform containing datasets for machine learning algorithms. The Credit Risk dataset contains more than thirty thousand records, each of which represents a separate loan that the client took from the bank. There are a total of 12 different signs that each entry (line) in the table contains. These features contain information about the customer and the loan he took (for example, the customer's age, his income, the amount of the loan, the interest rate, etc.). Among these attributes, there is the attribute "loan_status", that is, the status of the loan: this is a binary value that takes the value "0" if the client has paid off the loan, and "1" otherwise.

The set of signs is combined: among them are categorical and numerical. Since there are limitations to working with hybrid data and modern libraries do not support work with both hybrid and continuous BNs, all continuous data were discretized.

2.2. Structured learning of the BN

A BN structure for credit risk assessment was learned from the dataset using a machine learning algorithm that incorporated constraints related to the construction of graph arcs. Due to the lack of expert knowledge in the field of lending and credit risks, as well as the availability of a large set of data, it was decided to use the Hill Climb Search algorithm.

The task of learning the structure of a BN can be formulated as follows: Given a training dataset $D = \{v^1, \ldots, v^m\}$, here v^i is an instance from the dataset. It is necessary to find a directed acyclic graph G^*, which satisfies the condition:

$$G^* = \arg\max_{G \in G^n} f(G\colon D), \tag{1}$$

here $f(G\colon D)$ is an evaluation function (metric) used as a fitness measure of some directed acyclic graph G of the dataset D, and G^n is the set of all possible directed acyclic graphs with n variables.

Local search methods (especially Hill Climb Search) traverse the search space, starting from some initial solution and taking a finite number of steps. At each step, the algorithm considers only local changes and selects the one that leads to the greatest improvement in the score f. The algorithm stops when there are no local changes leading to improvement f. Due to this special behavior, the execution of the algorithm stops when only the local extremum is reached, not the global extremum. Various strategies are used to avoid this: restarts, randomness, etc.

When training BN for local changes in the space of directed acyclic graphs, such operations as adding an arc, removing an arc, and changing an arc are usually used. It is obvious that, except in the case of removing arcs, it is necessary to avoid creating directional loops. Thus, there are $O(n^2)$ possible changes, where n is the number of variables. As for the initial solution, the empty network is usually considered, although random starting points or perturbed local optima are also used, especially in the case of iterative local search.

Effective evaluation of directed acyclic graphs is based on an important property of evaluation metrics: decompositivity in the presence of complete data. For BNs, decompositive metrics evaluate a given acyclic graph as the sum of evaluations of the family of its nodes, that is, the subgraphs formed by some node and its parents in G. Formally, if f is decomposable, then:

$$f(G\colon D) = \sum_{i=1}^{n} f_D(X_i, Pa_G(X_i)), \tag{2}$$

$$f_D(X_i, Pa_G(X_i)) = f_D(X_i, Pa_G(X_i)\colon N_{x_i, pa_G(X_i)}), \tag{3}$$

where $N_{x_i, pa_G(X_i)}$ are variable statistics X_i and $Pa_G(X_i)$ in D, for example, the number of instances in D, which correspond to all possible instances X_i and $Pa_G(X_i)$.

Thus, if a decomposative metric is used, then a procedure that changes only one arc at each move can efficiently evaluate the subgraph produced by that change. The Hill Climb Search algorithm, which uses add, delete, and change arc operators, can take advantage of this mode of operation, and in particular needs to measure such differences when evaluating the improvement obtained by the subgraph:

a) adding an arc $X_j \to X_i$: $f_D(X_i, Pa(X_i) \cup \{X_j\}) - f_D(X_i, Pa(X_i))$;
b) removing the arc $X_j \to X_i$: $f_D(X_i, Pa(X_i) \backslash \{X_j\}) - f_D(X_i, Pa(X_i))$
c) changing the arc $X_j \to X_i$ performed as a deletion $(X_j \to X_i)$ plus addition $(X_i \to X_j)$,

so

$$[f_D(X_i, Pa(X_i) \backslash \{X_j\}) - f_D(X_i, Pa(X_i))] + [f_D(X_i, Pa(X_i) \cup \{X_j\}) - f_D(X_i, Pa(X_i))]. \tag{4}$$

Then, at each step, the algorithm analyzes all possible (local) operations and selects the one with the largest positive difference.

As a measure of evaluation of the constructed directed acyclic graphs, the $K2$ metric was used, which is calculated according to the formula:

$$K2(B,T) = \log(P(B)) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\log \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} + \sum_{k=1}^{r_i} \log(N_{ijk}!) \right), \tag{5}$$

where B is an optimal BN from the set of all possible networks, $B \in \mathrm{B}_n$; $T = \{y_1, \ldots, y_N\}$ is a training dataset; $P(B)$ is the joint probability distribution of the BN B; r_i is the number of states of the probability variable X_i; N_{ijk} is the number of instances in the dataset T, in which the variable X_i takes its k-th value x_{ik} and parent variables $Pa(X_i)$ are in the j-th configuration w_{ij}.

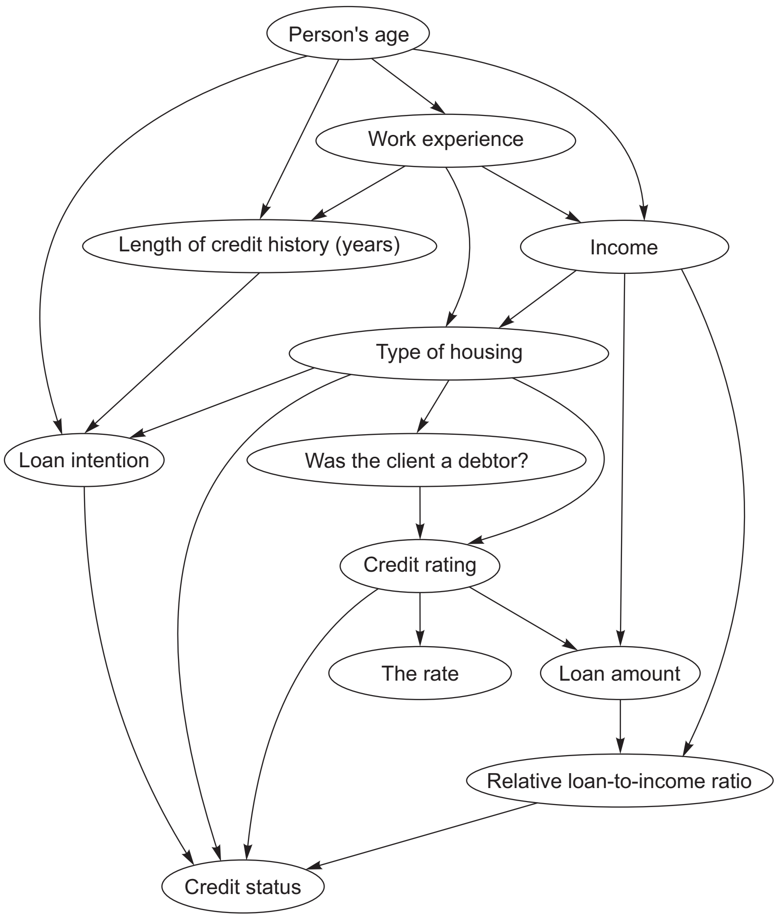

Fig. 1. The structure of a BN built on the basis of data on credit loans b.

In order to reduce the graph search space, the algorithm has been given input constraints, which are that it is forbidden to draw directed arcs from nodes of a lower level to any node belonging to a higher level, as well as some restrictions that follow from the logic of bank loan formation. By incorporating this knowledge into the Hill Climb Search algorithm from the pgmpy library and running this algorithm for execution, we will get a list of arcs that are the searched BN.

Visualization of the resulting graph using the GraphViz tool is shown in Figure 1.

2.3. Learning parameters

In the pgmpy library, it is possible to calculate the value of all conditional probabilities of a given BN, using Bayes estimation or maximum likelihood estimation (MLO). The first method uses an a priori distribution from the dataset, the second makes no initial assumptions. Since the Bayesian approach is inherently more stable and reliable, Bayesian estimation was chosen for parameter training.

In general, the Bayesian parameter estimation for a BN is as follows. Let the given structure G be a BN with parameters θ. Let the a priori distribution of all parameters of the BN also be established $P(\theta)$. Then the posterior distribution of the parameters with the given training dataset D is calculated as follows:

$$P(\theta|D) = \frac{P(D|\theta)\,P(\theta)}{P(D)}. \tag{6}$$

In the formula (6) $P(\theta)$ is a priori distribution, $P(D|\theta)$ is the probability of data for given values of unknown parameters, that is, the likelihood function, and $P(D)$ is a normalizing constant or the function of the isolated likelihood.

The likelihood function $P(D|\theta)$ can be decomposed into local likelihood functions:

$$P(D|\theta) = \prod_i L_i(\theta_{X_i|Pa_{X_i}} : D). \tag{7}$$

In addition, assuming global independence of parameters, we have

$$P(\theta) = \prod_i P(\theta_{X_i|Pa_{X_i}}). \tag{8}$$

Combining these two schedules, we obtain

$$P(\theta|D) = \frac{1}{P(D)} \prod_i \Big[L_i\big(\theta_{X_i|Pa_{X_i}} : D\big)\, P\big(\theta_{X_i|Pa_{X_i}}\big)\Big] \tag{9}$$

Several variants of prior distributions are available in the pgmpy library, and the equivalent Bayesian Dirichlet form was taken from these for training (BDeu). During the BDeu training process, for each variable, N uniformly distributed samples are generated to calculate the pseudo-frequencies (default $N = 5$), therefore, the estimated probabilities in the conditional distribution tables are more conservative than those obtained by MLO (meaning that probabilities close to 1 or 0 are smoothed out).

2.4. Results of model training

After learning the structure and parameters, the BN can be applied for statistical inference. In BNs, in order to learn the posterior distribution of a certain variable, not all the vertices of the network graph are needed, but only the parent, child vertices, and the parent vertices of the child vertices. This set is called a Markov cover, and pgmpy allows you to obtain a Markov cover for any node. For example, for the "loan_status" node, the markup coverage has the form showed in Figure 2.

Thus, in order to assess credit risks, that is, to calculate the probability that the client will not repay the loan, it is enough to know only 4 variables out of 11, namely: the intention of the loan, the ratio of the size of the loan to the client's income, the client's rating and the type of housing in which he lives client. Other variables will affect the posterior distribution only if at least one of the variables from the Markov coverage does not have a certificate.

By default, the pgmpy library uses the variable elimination algorithm for inference. The basis of the algorithm is the chain formula for the decomposition of the joint probability distribution of the network, which originates from Bayes' theorem:

$$P(X) = P\big(X^{(N)}|X^{(1)}, \ldots, X^{(N-1)}\big) \cdot P\big(X^{(N-1)}|X^{(1)}, \ldots, X^{(N-2)}\big) \cdot \ldots \cdot P\big(X^{(1)}\big)$$

$$= \prod_{i=1}^{N} P\big(X^{(i)}|X^{(1)}, \ldots, X^{(i-1)}\big) = \prod_{i=1}^{N} P\big(X^{(i)}|Pa\big(X^{(i)}\big)\big), \tag{10}$$

where $Pa\big(X^{(i)}\big)$ is the set of parent vertices relative to a vertex $X^{(i)}$.

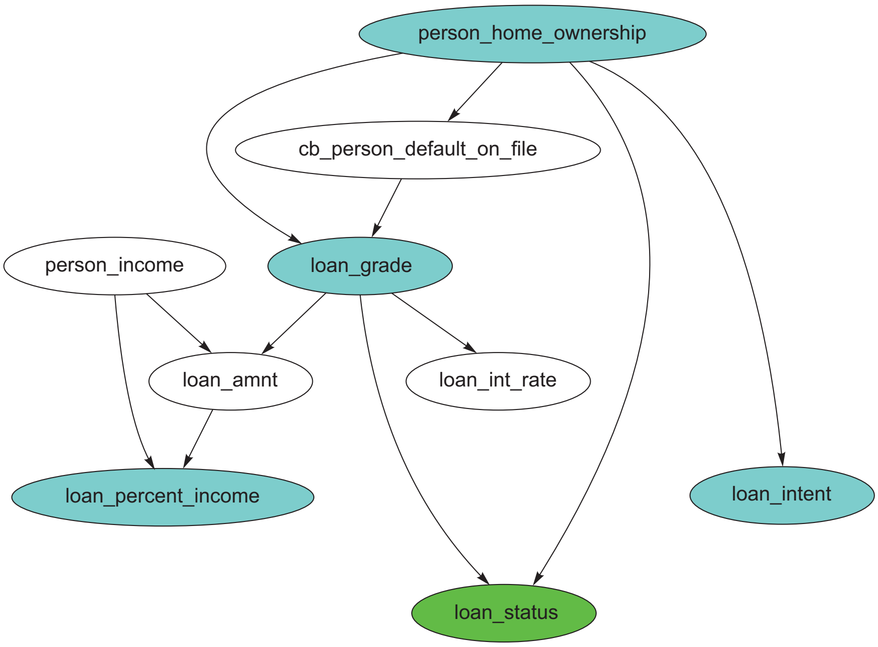

Fig. 2. Markov coverage for the loan_status variable is marked in blue.

The main idea of the variable exclusion algorithm is to calculate the probability of the top of the network using the formula:

$$P\big(X^{(n)}, e\big) = \sum_{X^{(k)}} \ldots \sum_{X^{(3)}} \sum_{X^{(2)}} \prod_{i} P\big(X^{(i)}|Pa\big(X^{(i)}\big)\big), \tag{11}$$

i.e., the less the vertex is connected to $X^{(n)}$, the farther it is outside the internal sum; each internal sum after calculation is converted into a new variable, which is further used as a multiplier.

Using the variable exclusion algorithm, a BN can be used to generate probabilistic queries against variables. Examples of such queries and the results of these queries:

— the general distribution among all clients of those who repay loans (value “0”) and those who do not repay them (value “1”);
— the probability that the client will and will not pay off the credit debt if he lives in a rented apartment;
— the probability that the client will and will not repay the loan if he lives in a rented apartment, but at the same time has the highest credit rating;
— distribution of loan sizes among customers who have paid off the loan and have average incomes.

The results of probabilistic inquiries can be directly used to assess credit risks. The information obtained as a result of the analysis can be used in the future to make decisions on issuing loans and to assess the solvency of certain categories of clients. Analysts can also use queries to any other variables for further analysis of the received statistical data.

3. Creation of a web application based on the BN

Using the Angular framework, a web application was created that visualizes a BN and allows interactive interaction with it, making inferences. The web application is built on the basis of client-server architecture. Angular was used to create a client interface (Figure 3) that interacts with the Flask server. Flask is responsible for processing requests, reading the BN model from a file and transferring data to the frontend.

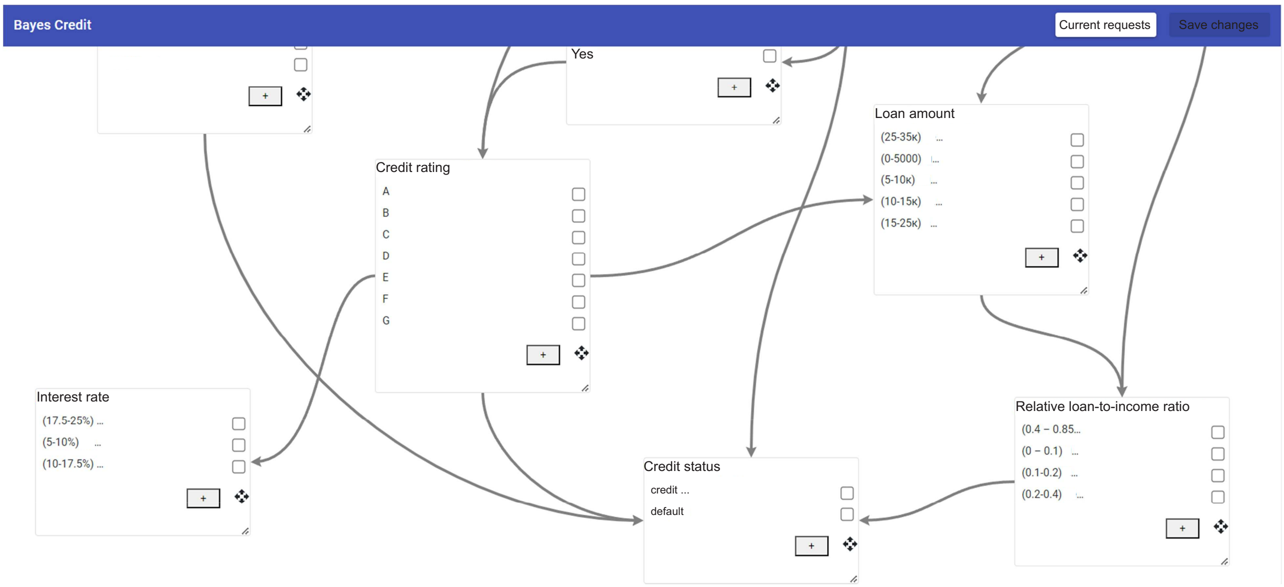

Fig. 3. The graphical interface of the web application based on the BN.

1. Frontend (Angular):
   — The frontend of the application is developed using the Angular framework, which provides high speed and efficiency.
   — The frontend is responsible for visualization and interaction with the user. The user can download or open a BN model, set certificates and receive credit risk analysis results through an intuitive graphical interface.
   — The frontend interacts with the backend by sending requests to download data about the BN, transfer data for analysis, and save the results.
2. Backend (Flask):
   — The backend is responsible for processing requests that come from the frontend. It reads a BN model from a file and processes it for further interaction.
   — The backend also performs inferences based on the established evidence, calculates the probabilities for the states of the specified variables and transmits the results of the calculations to the frontend.
3. Saving information about the BN:
   — A separate .json file stores information about the visual appearance of the BN graph, namely the order of nodes relative to each other.
4. The user interface:
   — The user interface is designed using HTML, CSS and Angular Material.
5. Communication between frontend and backend parts:
   — Frontend and backend interact via HTTP requests. The frontend sends requests to the server to load data, install certificates, and retrieve results. The backend processes these requests, performs the necessary operations and sends responses back.

The application allows us to visualize the inference of the BN and analyze possible credit risks accordingly. After launching the application, the user can send a request to the server to receive information about BN, which is used to assess credit risks. The server reads the BN model from the file and transfers the list of vertices, arcs and variable values to the client part. The graphical representation of the model (Figure 4) includes vertices, which represent different variables, and arcs, which reflect the dependencies between them. The user has the ability to set evidence for variables in BN to consider different scenarios and assess credit risks.

The user interface includes:

— BN nodes that can work in two modes: evidence variable and polled variable;

— button to move any node to any possible position, with the subsequent ability to save these variables. To do this, you need to click and hold the left mouse button on the corresponding icon;
— graph arcs that move synchronously with the movement of nodes;
— the "current queries" panel, in which the user can conveniently review all the surveyed variables (Figure 5).

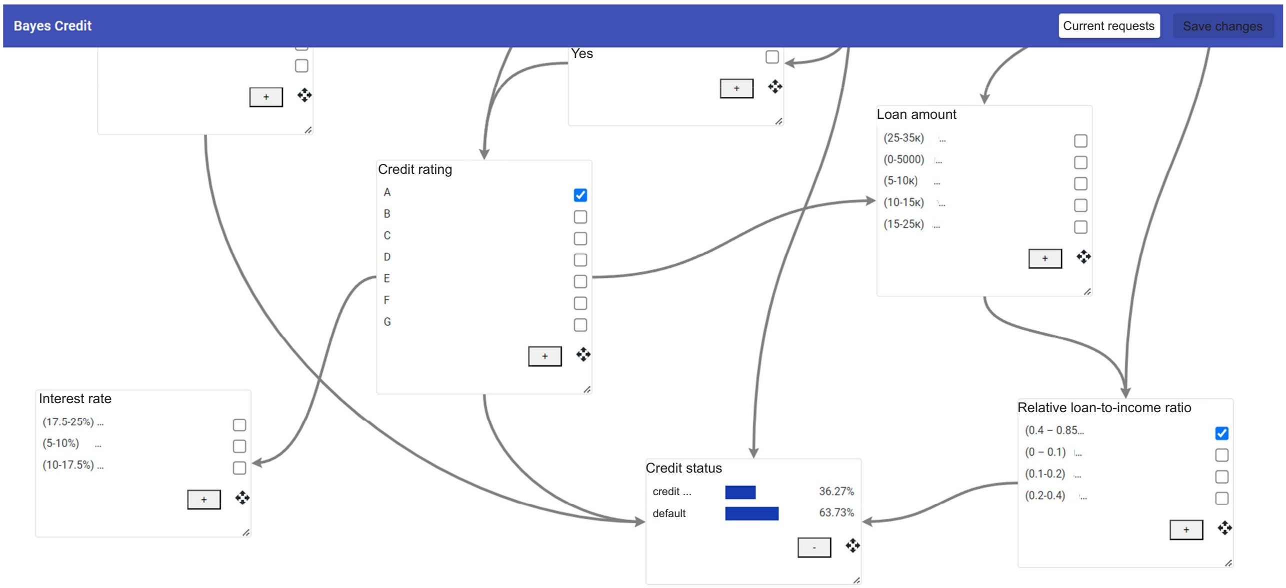

Fig. 4. A probabilistic query to a BN with established credentials.

Only one state can be selected for a certificate variable (using a checkbox). To make a probabilistic query to a variable, you need to press a button "+". In the polled variable mode, the user can see the probabilities for all states of the variable, taking into account all available evidence (if there is even one). As certificates change, the probabilities for all polled variables are recalculated automatically. To switch back to variable-certificate mode, you need to press the button "–".

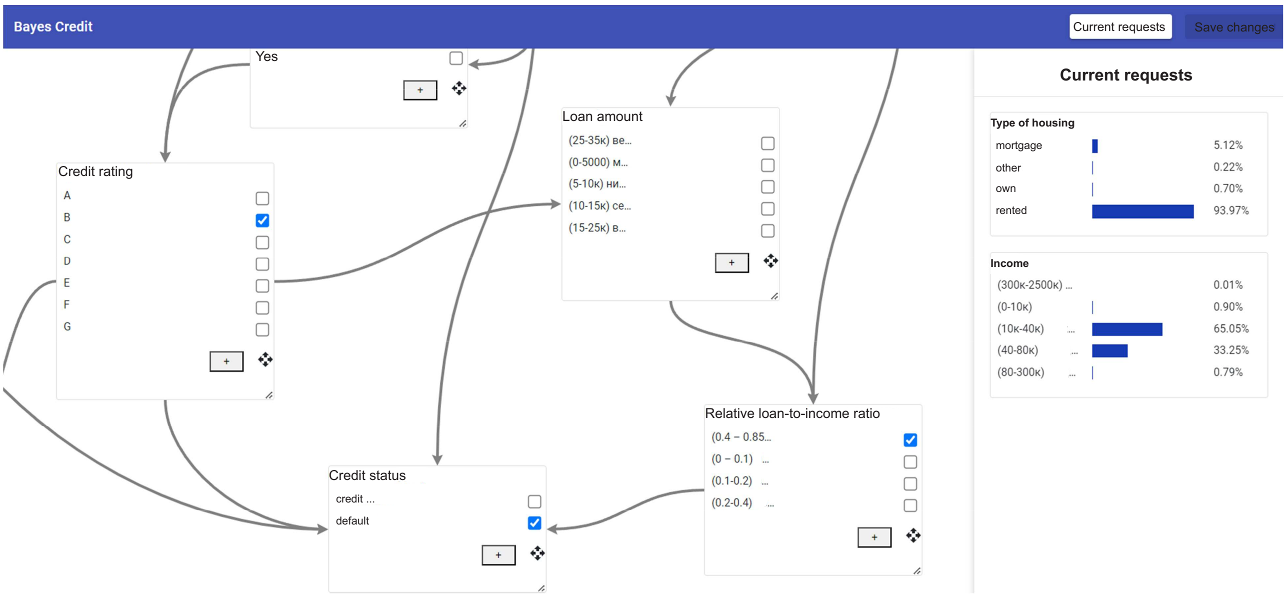

Fig. 5. A sidebar with all network nodes to which requests are made.

The main purpose of using BN in this application is to perform credit risk analysis based on available data and uncertainties. This web application provides opportunities to effectively manage loan portfolios and make informed decisions about granting loans. The use of BN allows us to take into account various factors that affect credit risks and conduct analysis based on real data. In addition, the web application can be a useful tool for customers applying for credit. They can use the app to assess the likelihood of getting a loan and identify potential risks.

Thanks to the capabilities of the web application, users can:

1. Model and analyze credit risks for various scenarios.
2. Determine the influence of various factors on credit risk.
3. Make informed decisions on granting loans based on probabilistic assessments.

The created web application illustrates the importance of using modern information technologies and innovative methods to achieve a better understanding and management of financial risks in the modern world. It helps analyze risks and improve the efficiency of the lending process, which is critical for financial institutions and their customers.

4. Conclusions

An overview of available software products and services was carried out for credit risk analysis and forecasting.

Modern information technologies and programming languages, particularly the Python programming language and the JupyterLab development environment, were used to analyze and pre-process the loan dataset. Based on the data, the structure of the BN was built using the machine learning algorithms available in the Pgmpy library. Model parameters were trained using machine learning algorithms.

Implementation of probabilistic queries to the developed BN and application of the built model for credit risk assessment is demonstrated. Using the Python programming language, the Flask web framework, and the Angular framework, a web application was created that visualizes the developed BN. The application allows you to interactively use the constructed mathematical model to calculate the probability of customers paying loans in the presence or absence of additional conditions.

[1] Ptak-Chmielewska A., Kopciuszewski P. New Definition of Default—Recalibration of Credit Risk Models Using Bayesian Approach. Risks. **10** (1), 16 (2022).

[2] Masmoudi K., Abid L., Masmoudi A. Credit risk modeling using Bayesian network with a latent variable. Expert Systems with Applications. **127**, 157–166 (2019).

[3] Baesens B., Rösch D., Scheule H. Bayesian Methods for Credit Risk Modeling. In: Credit Risk Analytics (eds B. Baesens, D. Rösch and H. Scheule), (2017).

[4] Leong C. K. Credit Risk Scoring with Bayesian Network Models. Computational Economics. **47** (3), 423–446 (2016).

[5] Lu J., Wu D., Dong J., Dolgui A. A decision support method for credit risk based on the dynamic Bayesian network. Industrial Management & Data Systems. **123** (12), 3053–3079 (2023).

[6] Kashmoola M. A., Aziz S. F., Qays H. M., Alsaleem N. Y. A. Unbalanced credit fraud modeling based on bagging and bayesian optimization. Eastern-European Journal of Enterprise Technologies. **3** (4 (123)), 47–53 (2023).

[7] Gmehling P., La Mura P. A Bayesian inference model for the credit rating scale. Journal of Risk Finance. **17** (4), 390–404 (2016).

[8] Tham A. W., Kakamu K., Liu S. Bayesian Statistics for Loan Default. Journal of Risk and Financial Management. **16** (3), 203 (2023).

[9] Wu J., Gao X. Quantification of Debt Default Based on Bayesian Model Averaging. ICBDT '23: Proceedings of the 2023 6th International Conference on Big Data Technologies. 86–95 (2023).

[10] Senyk A., Manziy O., Futryk Y., Stepanyuk O., Senyk Y. Information system supporting decision-making processes for forming of securities portfolio. Journal of Lviv Polytechnic National University "Information systems and networks". **11**, 39–55 (2022).

[11] Sverstiuk A., Dubynyak T., Manziy O., Senyk A., Ohloblin P. Bayesian click model and methods of estimating its parameters. CEUR Workshop Proceedings. **3628**, 389–403 (2023).

[12] Dubyniak T. S., Manziy O., Gancarczyk T., Senyk A., Futryk Y. Specialized information system for support of the process of recruiting securities. CEUR Workshop Proceedings. **3468**, 117–125 (2023).

[13] Arosio M., Martina M. The use of Graph Theory to improve disaster risk assessment. Geophysical Research Abstracts. **20**, EGU2018-18451 (2018).

[14] Shivraj V. L., Rajan M. A., Balamuralidhar P. A graph theory based generic risk assessment framework for internet of things (IoT). 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). 1–6 (2017).

[15] Bahar A. Y., Shorman S. M., Khder M. A., Quadir A. M., Almosawi S. A. Survey on Features and Comparisons of Programming Languages (PYTHON, JAVA, AND C#). 2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS). 154–163 (2022).

Застосування байєсівського підходу для моделювання кредитних ризиків

Сеник А. П.[1], Манзій О. С.[1], Оглоблін П. Є.[1], Сеник Ю. А.[2], Красюк О. П.[3]

[1] *Національний університет "Львівська політехніка", вул. С. Бандери, 12, 79013, Львів, Україна*
[2] *Національний лісотехнічний університет України, вул. Генерала Чупринки, 103, 79057, Львів, Україна*
[3] *Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, вул. Героїв Майдану, 32, 79026, Львів, Україна*

Розроблено комп'ютерну модель для аналізу, оцінювання та прогнозування банківських кредитних ризиків. На основі байєсової мережі та існуючих способів оцінки її параметрів створено програмний продукт (комп'ютерну модель) із використанням мови програмування Python, який здійснює прогнозування ймовірності того, що позичальник не виконає своїх фінансових зобов'язань, таких як, наприклад, погашення кредиту.

Ключові слова: *математичні методи; аналіз даних; графік; ймовірнісні параметри; машинне навчання; моделі кредитного ризику; байєсовська мережа; Байєсівський підхід.*