Electronic Research Archive

ERA, 32(8): 5139–5156.
DOI: 10.3934/era.2024237
Received: 19 December 2023
Revised: 17 May 2024
Accepted: 01 July 2024
Published: 30 August 2024

http://www.aimspress.com/journal/ERA

Research article

Graph-based two-level indicator system construction method for smart city information security risk assessment

Li Yang[1], Kai Zou[1] and Yuxuan Zou[2,*]

[1] School of Public Administration, Xiangtan University, Xiangtan 411105, China
[2] Schools of Management, Xi'an Jiantong University, Xi'an 710049, China

* **Correspondence:** Email: yuxuan.zou@foxmail.com.

Abstract: The rapid development of urban informatization has led to a deep integration of advanced information technology into urban life. Many decision-makers are starting to alleviate the adverse effects of this informatization process through risk assessment. However, existing methods cannot effectively analyze internal and hierarchical relationships because of the excessive number of indicators. Thus, it is necessary to construct an indicator's dependency graph and conduct a comprehensive hierarchical analysis to solve this problem. In this study, we proposed a graph-based two-level indicator system construction method. First, a random forest was used to extract the indicators' dependency graph from missing data. Then, spectral clustering was used to separate the graph and form a functional subgraph. Finally, PageRank was used to calculate the prioritization for each subgraph's indicator, and the two-level indicator system was established. To verify the performance, we took China's 25 smart cities as examples. For the simulation of risk level prediction, we compared our method with some machine learning algorithms, such as ridge regression, Lasso regression, support vector regression, decision trees, and multi-layer perceptron. Results showed that the two-level indicator system is superior to the general indicator system for risk assessment.

Keywords: smart city; information security; risk assessment; two-level indicator system construction; machine learning

1. Introduction

The application of advanced information technologies such as the Internet of Things, cloud

computing, and big data have profoundly affected the development pattern of cities. While people enjoy the convenience they bring, they also face increasingly hazardous information security problems, such as virus flooding, hacker attacks, and network fraud. Driven by economic, political, military, and other interests or even due to non-malicious events such as technical flaunts and pranks, attacks such as information theft, tampering, and destruction occasionally occur. Attack methods vary, and information attacks have evolved from traditional external attacks to internal ones, forming a trend of combined internal and external threats [1].

For a long time, information security has been of great interest, being studied in many fields such as politics, business, and academia; more and more researchers have focused on the risk assessment and management of information security [2–4]. In the process of urban development, information security problems also exist and gradually increase with the development of urban information. The construction of a smart city is a complex system project, and factors (risk indicators) such as technology development and application of smart devices, communication between devices, software platform security, data storage and encryption, and people's information security education are all issues that need to be considered. Many cities blindly carry out the construction of smart cities without evaluating their real needs and without overall planning; this results in the construction of smart cities that seem to be prosperous, but, in fact, have a chaotic internal management. Thus, there is an urgent need to construct indicator systems that can clearly distinguish the functions of each information security risk indicator. However, due to the excessive number of indicators involved, these functions are messy and indistinguishable, and the construction of an indicator system consistently encounters difficulties. Therefore, this paper evaluates a method for constructing a two-level indicator system of information security risk assessment based on graphs and tries to explore related issues to provide a basis for information security decisions.

Risk assessment is the foundation of information security management, providing theoretical support for the protection of critical information assets and the avoidance of security risks. Information security risk assessment requires actively identifying information security risks, studying the basic elements (indicators) of such information security risks, quantifying them, and balancing decision-making behaviors between the assets to be protected and the costs. The so-called information security risk refers to information danger and loss and its impact on the organization, caused by a security incident due to a system vulnerability, either man-made or due to natural threats.

Information security risks are characterized by randomness, fuzziness, and uncertainty, which makes it difficult to establish mathematical models to analyze them. In 1998, Finne [5] proposed a conceptual model that clearly pointed out that information security risks are a function of information assets, threats, and vulnerabilities, being defined by the possibility of and potential for loss of information assets caused by threats or vulnerabilities. Information security risk assessment is a comprehensive assessment of threats, vulnerabilities, and information assets. Most information security risk assessments since then have followed this line of thinking.

From the literature analysis, information security risk assessment methods can be summarized into the following categories:

1) Fault tree–based hazard analysis. Herzog and Shahmehri [6] used fault tree analysis (FTA) to investigate the harmful events of information security risks. They used the monitor and control system (MCS) combination and transmission path of all minimum cut sets of the fault tree to identify events and their consequences and find out the sets of factors leading to information risks, using the structure function in the form of minimum cut set to describe the fault tree. Zhu et al. [7] proposed a

new risk assessment model based on the belief rule base (BRB) system and FTA, which establishes FTA rules based on the BRB and expands the knowledge base through the FTA algorithm. In addition, the model is optimized to reduce the uncertainty in the model. However, the calculation of the fault tree is complicated and is not suitable for information security risk assessment with many security events.

2) Fuzzy comprehensive evaluation method [8,9]. In this field, fuzzy theory and analytic hierarchy processes have gained some interest. Most literature focuses on the analytic hierarchy process, fuzzy comprehensive evaluation method, and their combination. The main idea is to organize the information system and risk impact hierarchically, establish corresponding evaluation indicators, determine the weight of each indicator through an analytic hierarchy process, and adopt a multi-attribute decision-making method to comprehensively evaluate the risk of the information system.

3) Knowledge-based information security risk assessment [10,11]. This mainly relies on the experience gained from security experts to solve the risk assessment problem of similar scenarios. The advantage of this approach is that it can directly provide recommended protection measures, structural frameworks, and implementation plans.

4) Model-based information security risk assessment. The model-based method can model all risk factors in the internal mechanism of the information system and all abnormal or harmful behaviors between the system and the external environment to complete the qualitative and quantitative analysis of the system's vulnerability and security threats [12–16]. A typical approach is CORAS [17], which provides a way to use case diagrams in UML and their extensions (improper use case diagrams) for risk analysis. In this approach, malicious or misused behaviors that could compromise the benefits and security of the system or other actors are modeled using improper use case diagrams. Similarly, Alfakeeh et al. [18] used the hesitant fuzzy-based AHP-TOPSIS technique to estimate the risks of various web applications for improving security durability. This approach would help to design and incorporate security features in web applications that would then be able to battle threats on their own.

In addition, many neural network algorithms are also used in this field due to their powerful nonlinear processing and learning abilities [19]. In 2021, Song and Xu [20] proposed a PSO-BPNN (particle swarm optimization-BPNN) model for information security risk assessment. In this method, PSO was used to find the best initial value before network iteration to address the slow convergence and accuracy problems of BPNN. In summary, although the aforementioned methods have achieved excellent results, the widespread use of neural networks still poses a huge challenge because of the black box problem (i.e., unclear intermediate process).

According to the above literature, a certain research progress has been made. However, it should be noted that due to the comprehensive effect of technology, governance, manpower, and external economic, social, ecological, and other factors, the construction of smart cities still faces a large number of complex problems. Among the existing research results, most studies only consider the perspective of information technology, and few analyze multiple perspectives. In addition, the determination of information security risks rarely considers the ambiguity of information risk indicators in the decision-making process and the uncertainty, both the extral uncertainty when experts make decisions and the internal uncertainty when experts make decisions on the same target at different times. While several existing approaches aim to resolve such ambiguity and uncertainty, more and more indicators are added to information security risks as the reality becomes more complex, which greatly deepens the impact of ambiguity and uncertainty.

This paper believes that one of the external reasons for the low accuracy of smart city information

security risk assessments is that most methods can only treat each risk indicator equally and cannot analyze the internal relationship between them. Although a number of approaches have recently emerged to utilize the two-level indicator body system, they all require extensive industry experience. In addition, obtaining relevant data is difficult. Therefore, we propose a method to construct a two-level indicator system of information security risk assessment based on graphs. First, the dependent network among indicators is constructed using the random forest algorithm to overcome the indicator uncertainty brought by data. Then, based on spectral clustering and the PageRank algorithm, the network is separated and the important relationships among the subgraphs are investigated. Finally, an adaptive multi-layer indicator system is constructed, which provides a way to clearly divide indicator relations and overcome fuzziness and uncertainty.

2. Methods

Information security risk assessment is the process of assessing the security attributes (e.g., confidentiality, integrity, and availability) of information systems and the information they process, transmit, and store. The construction of an information security risk assessment indicator system aims to predict possible risks and put forward corresponding solutions. However, the number of indicators multiplies with the expansion of the urban information security risk system, and the dependence relationship among those indicators becomes complicated. How to identify the core evaluation indicator from the complex system is the key task. In this paper, a graph-based two-level risk assessment indicator system construction method is proposed (Figure 1). In this method, the risk indicator is taken as the node, and the relationship between the indicators is taken as the edge to build a graph (network) of indicator relationships. Specifically, the method first uses a random forest to extract the interactive network of indicators from the dataset; then, the subgraph for the construction of the two-level indicator system is obtained. Next, the PageRank algorithm is used to search the core indicators' prioritization and analyze the two-level indicator system. Finally, taking China's 25 smart cities as examples, the general indicator system and the two-level indicator system are input into machine learning algorithms such as ridge regression (Ridge), Lasso regression (Lasso), support vector regression (SVR), decision trees, and multi-layer perceptron (MLP) for a simulation application, and results are obtained.

2.1. Random forest regression-based indicator dependency network construction

In the construction of the indicator system of a smart city, the main difficulties are the lack of system explainability and the curse of dimensionality caused by a large number of indicators as well as the ambiguity and uncertainty brought by missing data. To eliminate the influence of these factors on the results as much as possible, the random forest regression algorithm is used to read the original data when obtaining key characteristic variables for subsequent processing.

The random forest [21] is a kind of ensemble learning, which can perform data prediction by integrating multiple decision trees. The most important advantages of the random forest applied to indicator system construction are as follows: First, it is a bootstrap sampling method, which makes it more robust against missing and unbalanced data. Second, it randomly selects features to divide data space, making it insensitive to multivariate collinearity. From a mathematical point of view, if the risk level of a city is regarded as a linear combination of all risk indicators, then the multivariate collinearity problem is reflected

as a fuzzy problem among indicators; that is, since each variable cannot be accurately distinguished, one indicator is linearly represented by another indicator. Third, it can calculate the importance of variables (usually via the Gini coefficient and minimum variance), which makes it a good method for clarifying the role of all indicators in the data. These three points make up for the most common errors in the construction of a smart city indicator system. Therefore, it has unique advantages for extracting indicator dependency networks from origin data. The basic steps of random forest regression are as follows:

1) Bootstrap sampling: Let $D = \{X; Y\}$ be the origin dataset, which includes n samples X and n real values Y. Then, m (m < n) samples and their real values are randomly extracted from D, and the sub-dataset $D_{sub} = \{X = (x_1, x_2, \cdots, x_m), Y = (y_1, y_2, \cdots, y_m)\}$ is constructed (repeat K times to obtain K sub-datasets).

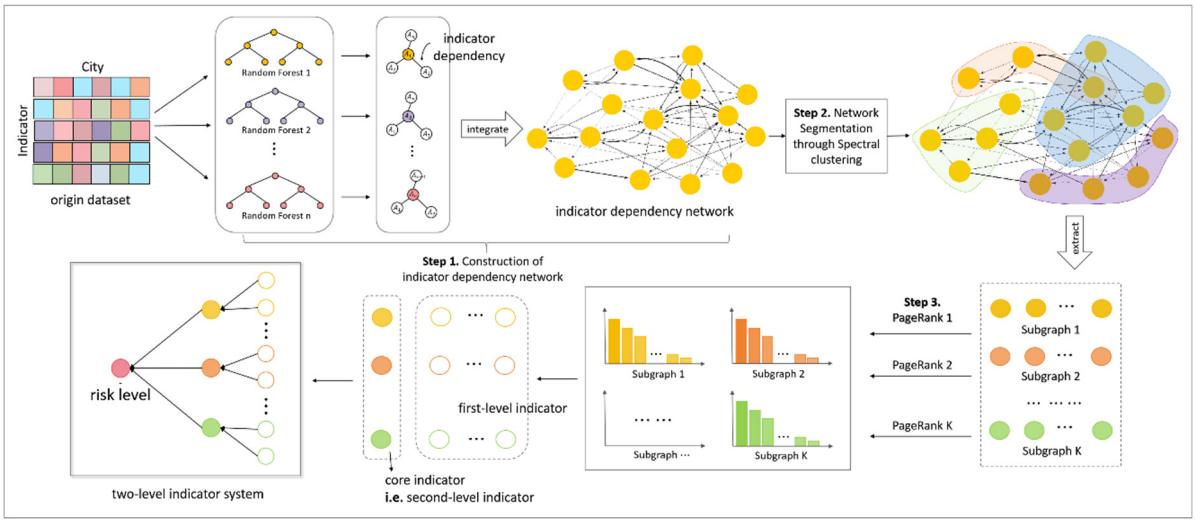

Figure 1. The flowchart of the information security risk assessment. Step 1. Construction of an indicator dependency network. There are 16 random forests in total, and each random forest constructs a dependency relationship between an indicator and other indicators. Step 2. Network segmentation. The network is divided into K subgraphs by spectral clustering. Step 3. Second-level indicator search. PageRank is run on K subgraphs to find the core indicator and take it as the second-level indicator. Finally, the two-level indicator system is constructed.

2) Training decision tree: Training K decision trees based on K sub-datasets; the minimum variance is used to determine the optimal segmentation variable f_{opt} and the optimal segmentation point e_{opt}, and the optimal segmentation variable is used as the optimal feature to construct the decision tree. The calculation method of minimum variance is as follows:

$$\min_{f_{opt}, e_{opt}} [\min_{c_1} \sum_{y_i \in f(R_1)} (y_i - c_1)^2 + \min_{c_1} \sum_{y_i \in f(R_2)} (y_i - c_2)^2] \tag{1}$$

$$R_1 = \{x | x^{(f_{opt})} \leq e_{opt}\}, R_2 = \{x | x^{(f_{opt})} > e_{opt}\} \tag{2}$$

$$c_m = \frac{1}{|R_m|} \sum_{y_i \in f(R_m)} y_i, m = 1,2 \tag{3}$$

where $f(\cdot)$ indicates that data are mapped to the corresponding y value in the dataset. R_1 and R_2 are the result of data segmentation when the current partition variable and partition point are used as the optimal value. The implied operation in formula (2) is that every time the variable f_{opt} is used as the segmentation variable, the value of the data on this feature needs to be sorted in ascending order before e_{opt} is selected.

3) Average vote: K decision trees are combined to form a random forest, and the mean prediction results of K decision trees are returned as the prediction results of the random forest.

The index system is too complicated, and the relationship between indicators is ambiguous. Constructing an indicator dependency graph can help us deal with and evaluate the indicator system visually and conveniently support decision-making. The idea adopted in this paper is to build the indicator dependency graph based on the indicator importance obtained by the random forest. The details are as follows:

1) Each indicator in the original data is taken as a linear combination of other indicators and, in turn, is fit by the random forest.

2) According to the characteristics of the random forest, the importance of other indicators to the target indicator is obtained.

3) The importance of all indicators to other indicators is regarded as the relative weight between them, and the indicator dependency graph is constructed.

2.2. Clinical trial registration

Although the existing indicator relationship network can express the dependence of each indicator, it cannot reflect the importance degree or type of the indicator. So far, all indicators in the network are equal, collectively referred to as first-level indicators. In this section, we need to separate these indicators into different categories and levels and identify second-level indicators. Graph-based clustering algorithms can be used to separate graphs to achieve the effect of indicator classification. In this paper, a spectral clustering algorithm is used to cluster indicators to realize the automatic segmentation of indicator dependency networks [22].

Spectral clustering divides weighted, undirected graphs into two or more optimal subgraphs so that the internal subgraphs are as similar as possible and the distance between subgraphs is as far as possible. The use of spectral clustering to condense indicators has the following advantages: First, clustering based on dependency graphs can simultaneously consider the whole indicator system rather than the relationship between two indicators. Second, according to the idea of graph segmentation, the obtained indicators are grouped into many independent modules, which is convenient for decision-makers to clearly understand the functions of indicators and the boundaries between indicators. The algorithm flow is as follows:

1) Build an adjacency matrix $W = [w_{ij}]_{n\times n}$ and degree matrix $D = diag(d_1, d_2, \cdots, d_n)$ from the indicator dependency network, where $d_i = \sum_{j=1}^{n} w_{ij}$;

2) Construct the standardized Laplacian matrix $\tilde{L} = D^{-1/2}LD^{-1/2}$, where $L = D - W$;

3) Obtain eigenvector matrix F from $\tilde{L}$ and normalize it;

4) Take each row in F as the data of sample (indicators) and put it into the K-means algorithm;

5) Obtain clusters $Subgraph_1, Subgraph_2, \cdots, Subgraph_K$.

2.3. PageRank-based two-level indicator system construction

The cluster obtained through spectral clustering cannot determine the cluster center, consequently preventing us from determining the key indicator of the divided sub-indicator system. Depending on the PageRank algorithm and the indicator dependency graph, we can determine the most critical indicator in the indicator system. In this paper, we use the PageRank algorithm to obtain indicators' prioritization for each subgraph obtained through spectral clustering as a substitute for secondary indicators.

The PageRank algorithm is a representative algorithm of graph link analysis. It was originally used as a calculation method for the importance of internet pages and in page ranking of the Google search engine [23]. The basic principle is the first-order Markov chain, which describes the behavior of random visits to nodes along the digraph. Under certain conditions, the probability of visiting each node will converge to the stationary distribution. Then, the stationary probability value of each node is its PageRank value: the higher the PageRank value, the more important the webpage. The main steps of the PageRank algorithm are as follows:

1) Build an adjacency matrix $W = [w_{ij}]_{n\times n}$ based on weighted undirected graphs.

2) Obtain the transition probability matrix by normalizing W so that the transition matrix has the following properties:

$$\sum_{j=1}^{n} w_{ij} = 1\,, i = 1,2,\cdots,n \tag{4}$$

$$0 \le w_{ij} \le 1 \tag{5}$$

3) Randomly initialize a unit vector $P^{(0)} = [p_{ij}]_{1\times n}$;

4) Iteratively simulate the Markov chain

$$P^{(t+1)} = P^{(t)}W \tag{6}$$

Once $t \to \infty$, $P^{(t+1)}$ will converge to a stable distribution at which point $P^{(t+1)}$ describes the PageRank value of each indicator.

For each subgraph segmented in the previous step, we conducted PageRank on them to search for key indicators. Finally, based on the indicator prioritization and subgraph, the following formula is used to construct the two-level indicators (imaginary) and their data:

$$Ind2nd(i) = \sum_{ind\in SubGrap\square_i}\big(1 - PageRank(ind)\big) * X(ind), i = 1,2,\cdots,K \tag{7}$$

where $Ind2nd(i)$ represents the expression data of second-level indicators. Ind is the indicator in the four subgraphs, and $PageRank(ind)$ and $X(ind)$ are the PageRank value of the indicator and the corresponding expression data, respectively. In this formula, $PageRank(ind)$ is not used as the coefficient instead of $1 - PageRank(ind)$ because PageRank selects key vertices based on the maximum entry degree or the maximum weight, which means that these vertices are highly likely to be calculated by other vertices. Therefore, multicollinearity, which will affect the final result, is a potential risk, and we should weaken it.

3. Case study

3.1. Problem description

As for the construction of a smart city information security indicator system, few current practices exist to convert it into a graph. In this paper, the original data were input, and the dependency graph among indicators was constructed by calculating the feature importance in the random forest. On this basis, the construction of the indicator system was converted into the field of graph analysis. Subsequently, spectral clustering and PageRank were used to analyze and obtain the two-level indicator system. The advantages of doing so are as follows: First, the expression of the indicator system is more intuitive, and the relationship between all indicators can be clearly understood. Second, the system facilitates systematic analysis of all indicators. Third, it has strong extensibility, convenient for further inferring the direction of indicators (causality).

To verify the performance of the method, 25 smart cities in China were selected for information security risk assessment. Meanwhile, to avoid excessive data loss due to the low degree of urban intelligence, four levels were selected: 4 first-tier cities, 11 new first-tier cities, 9 second-tier cities, and 1 third-tier city (Supplementary material 1). Table 1 lists the risk indicators. A1–A16 are risk indicators, covering the four categories of people, platform, policy, and data. A17 is the city's comprehensive risk level as Y in the method.

Table 1. Description of the indicators.

A1	Communication network construction	A10	Data encryption and recovery
A2	Network resource connection	A11	Data backup technology
A3	Urban cloud platform construction	A12	Data opening service level
A4	Legitimacy of information content	A13	Research and development spending
A5	Authenticity of information content	A14	Firewall reliability
A6	Controllability of information content	A15	Operating system security
A7	Safety education and training	A16	Vulnerability threat repair rate
A8	Safety knowledge promotion	A17	Comprehensive development level of the city
A9	Public safety consciousness		

3.2. Information security risk assessment

3.2.1. Random forest regression-based indicator dependency network construction

To construct the interactive network of indicators, we take each indicator in A1–A16 as y and others as x to construct a set of regression equations

$$\begin{cases} A_1 = f_1(A_2, A_3, \cdots, A_{16}) \\ A_2 = f_2(A_1, A_3, \cdots, A_{16}) \\ \cdots\cdots \\ A_{16} = f_{16}(A_1, A_2, \cdots, A_{15}) \end{cases} \tag{8}$$

where each equation in the system describes the relationship between the target indicator and other indicators and is nonlinear and extensive (we do not assume that it satisfies any fixed form). However, mathematically solving it is nearly impossible. Fortunately, random forests can build tree mappings from the data domain (X) to the value domain (Y). Accordingly, we trained random forests to preserve these equations in a structured form instead of a mathematical formula.

The training of this model involves the setting of an important parameter *n_estimators* (i.e., the number of decision trees). We test the performance of this parameter within a range of 1–50 according to the $FitScore$ and select the largest one as the final parameter:

$$FitScore = Normalisation(R^2) - Normalisation(MSE) \tag{9}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{11}$$

where $FitScore$ describes how well the model fits the data, R^2 is the coefficient of determination, MSE is the mean square error, and $Normalisation(\cdot)$ is a normalization operation. The combination of R^2 and MSE can be used to measure the degree to which the model can be interpreted for variables whilst improving its accuracy. The indicators for the 50 different scenarios are as follows (Figure 2):

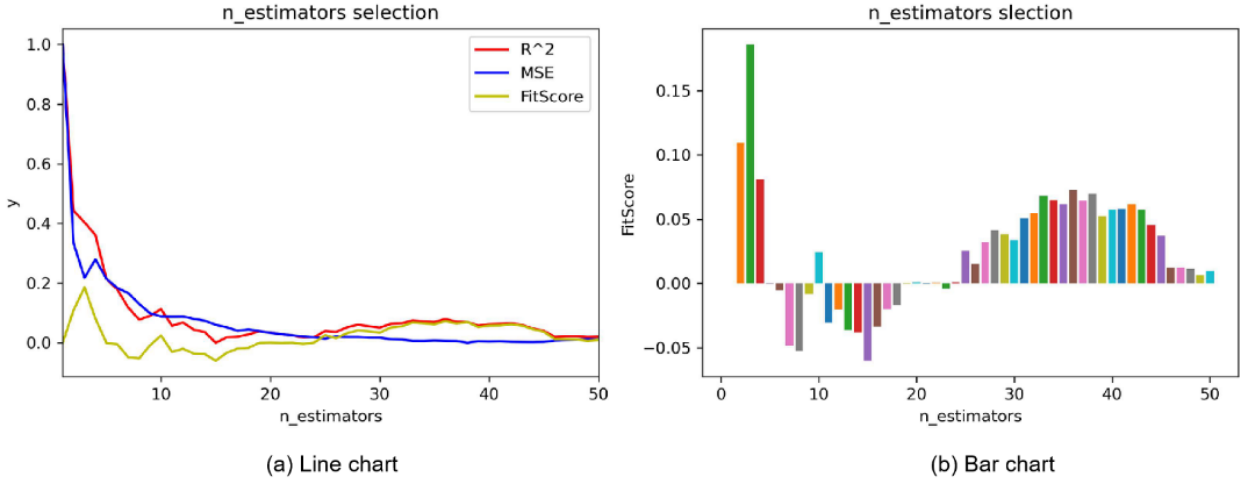

Figure 2. FitScore of the parameter *n_estimators* in the range of 1–50. (a) Records the value of MSE, R^2, and FitScore; (b) shows the FitScore in the form of a bar chart. The higher the FitScore, the better the result.

As can be seen from the figure, the optimal parameter should be *n_estimators*=3 ($FitScore = 0$ when *n_estimators=1*). Therefore, the number of decision trees in this paper is 3 to train 16 equations.

In the second step, based on the trained model, nodes of the tree are retrieved to obtain the contribution of each indicator to the target (the indicator has no contribution to itself). Finally, the contribution between 16 groups of indicators is used to construct the weighted dependency graph between indicators (Figure 3).

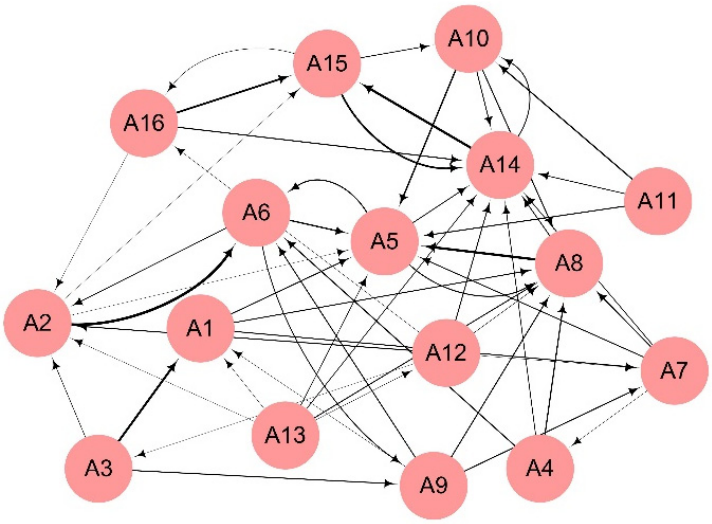

Figure 3. Weight dependency graph between indicators. Only edges larger than the average value are displayed, and the thicker the line, the greater the weight of the edge.

3.2.2. Random forest regression-based indicator dependency network construction

We obtained an indicator dependency graph that contains the relationships between the indicators. For the time being, all vertices in the graph are regarded as first-level indicators, and there is no difference in their classification and function. To construct a two-level indicator system, a direct idea is to divide indicators into multiple categories according to their roles or properties and then select an indicator from each category as a second-level indicator or generate a new indicator as a second-level indicator. In this study, we choose the latter and then take the first-level indicators under this category as sub-indicators of the second-level indicator.

On the premise of an input network graph, the most suitable way to classify vertices is network segmentation. By the network segmentation algorithm, the indicator dependency graph can be divided into several subgraphs. The vertices in different subgraphs are different categories of indicators. There are many network segmentation algorithms to choose from. After considering the segmentation effect and algorithm difficulty comprehensively, we choose a spectral clustering algorithm (for details, refer to Section 2.2). Moreover, since the input required by the algorithm is an undirected graph, and the indicator dependency graph is a directed graph, we have modified it. Let W be the weight matrix of the indicator dependency graph and W^T be the transposed matrix. Then, the input of spectral clustering is

$$\widetilde{W} = \frac{W+W^T}{2} \tag{12}$$

Finally, the results of spectral clustering are as follows: the indicator dependency graph is di vided into four independent subgraphs according to the preset clustering parameter K = 4. They are $SubGraph_1 = [\mathrm{A14}, \mathrm{A15}, \mathrm{A16}]$, $SubGraph_2 = [\mathrm{A1}, \mathrm{A2}, \mathrm{A3}, \mathrm{A6}]$, $SubGraph_3 = [\mathrm{A4}, \mathrm{A7}, \mathrm{A9}, \mathrm{A13}]$, and $SubGraph_4 = [\mathrm{A5}, \mathrm{A8}, \mathrm{A10}, \mathrm{A11}, \mathrm{A12}]$.

Take $SubGraph_1 = [\mathrm{A14}, \mathrm{A15}, \mathrm{A16}]$ and $SubGraph_2 = [\mathrm{A1}, \mathrm{A2}, \mathrm{A3}, \mathrm{A6}]$, for example. In $SubGraph_1$, fire reliability (A14), operating system security (A15), and vulnerability threat repair

rate (A16) reflect the security of the platform. The security of the platform can be taken as a second-level indicator, and this is consistent with reality. In a computer platform, security is affected by both internal and external aspects. From the outside, computer viruses are often wrapped in normal data to enter the system deceptively, and the reliability of the firewall is a layer of protection to prevent information security problems. Internally, whether there is a backdoor inside the operating system or there are loopholes in the system code, the system's defense against information security attacks and the repair rate after the attack are also important factors affecting information security. In $SubGraph_2$, communication network construction (A1), network resource connection (A2), and urban cloud platform construction (A3) control the generation and transmission of information from the perspective of the system and screen the information content layer upon layer to ensure the content security of urban information. This is also consistent with the facts.

Finally, with the analysis of $SubGraph_3$ and $SubGraph_4$, we conclude that these four subgraphs correspond to four aspects that affect information security, thus determining the basic composition of the two-level indicator system: platform security, information authenticity, information controllability, and information legitimacy. All 16 indicators work together to ensure the functions of these four aspects, and ultimately act on the whole indicator system, affecting the level of information security in the city.

3.2.3. PageRank-based two-level indicator system construction

The indicator subset obtained through spectral clustering can somewhat represent the second-level indicator system, but the core indicator of the four subsystems needs to be further determined to associate with the final urban information security risk.

The content of second-level indicators has been preliminarily determined through analysis in the previous section, so the content of this section focuses on obtaining the data of second-level indicators because those are generated rather than selected from the existing 16 indicators. A feasible method is to sum the data of all first-level indicators in the subgraph by weight as the data of the second-level indicators; the weight is determined by the PageRank algorithm. For a subgraph (containing m first-level indicators and their dependencies), let the weight matrix of the subgraph be represented as W_i; then, we can randomly initialize a unit vector $P^{(0)} = [p_{ij}]_{1\times m}$ and input $P^{(0)}$ and W_i to iterate through formula (6). Finally, we can obtain a stable $P^{(t+1)}$, where each component represents the PageRank value (i.e., weight) of each indicator in the subgraph (see Section 2.3 for details). PageRank results of indicators in the four subgraphs are as follows: Figure 4 shows that the core indicators corresponding to $SubGraph_1$, $SuGraph_2$, $SubGraph_3$, and $SubGraph_4$ are the security of the operating system (A15), the authenticity of the information content (A5), the controllability of the information content (A6), and the public safety consciousness (A9). This result is highly consistent with the clustering result shown in Figure 5. The four core indicators selected by PageRank are all the vertices with the highest degree or the most edges with high weight in the subgraph where they are located.

Finally, according to Eq (7), four imaginary second-level indicators and their data are extracted as second-level indicator entities.

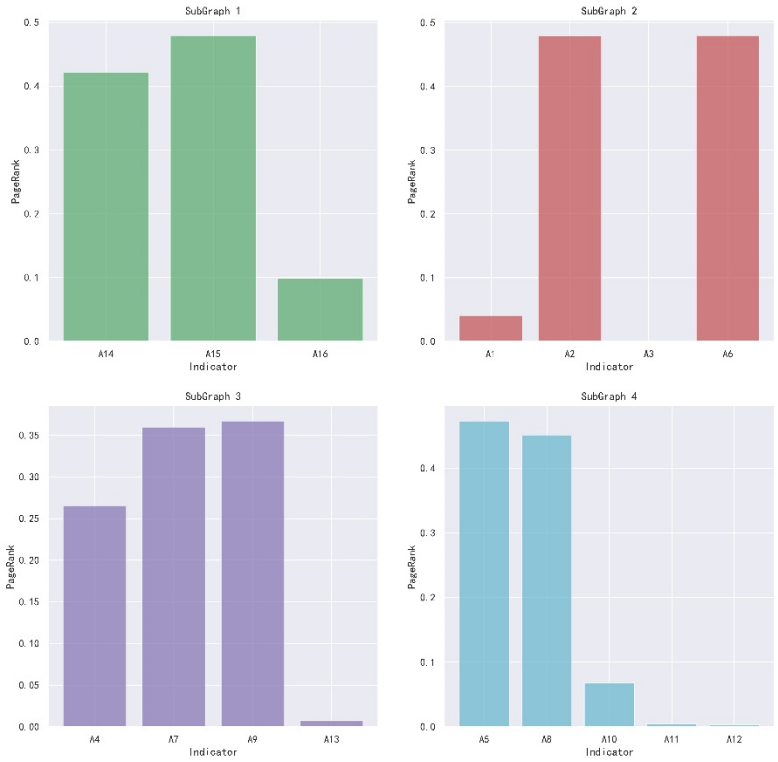

Figure 4. PageRank results of the four subgraphs. For each indicator, the higher the PageRank value, the more important it is in the subgraph.

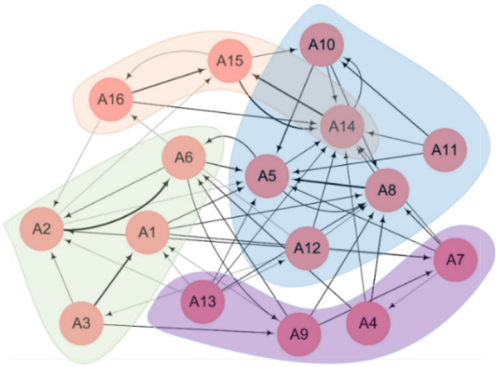

Figure 5. Spectral clustering results.

4. Discussion

Our goal is to prove that the two-level indicator system is superior to the single-level indicator system (the indicator set without any processing). Since we cannot apply it to the actual scene and verify the results, we treat it as a regression problem. Moreover, to verify the wide applicability of the indicator system we constructed, we use classical machine learning algorithms, such as Ridge, Lasso, SVR, decision tree (Dtree), and MLP for the experiment, instead of using some specific methods.

Specifically, we use the data from the two-level indicator system (including

$Ind2nd(1), Ind2nd(2), Ind2nd(3)$, and $Ind2nd(4)$) and the data from the single-level indicator system (including A1–A16) to fit the risk level of the city (as a regression problem):

$$y = f(Ind2nd(1), Ind2nd(2), Ind2nd(3), Ind2nd(4)) \quad (13)$$

$$y = f(A1, A2, \cdots, A16) \quad (14)$$

Then, Ridge, Lasso, SVR, Dtree, and MLP algorithms were fitted to Eqs (13) and (14) to verify the effect. The evaluation indicators used were R^2 and MSE (Equations 10 and 11). The experimental results are as follows (Table 2):

Table 2. Fitting performance of the single-level and two-level indicator systems.

Method	Single-level indicator system		Two-level indicator system	
	MSE	R^2	MSE	R^2
Ridge	**0.0041**	**0.8924**	0.0085	0.7749
Lasso	0.0369	0.0301	**0.0366**	**0.0389**
SVR	0.01	0.84	**0.01**	**0.85**
Dtree	0.0	1.0	**0.0**	**1.0**
MLP	0.0333	0.1263	**0.0214**	**0.4365**

Note: Bold values mark the algorithm's best performance in the single-level or two-level indicator system.

As can be seen from Table 2 and Figure 6, when the two-level indicator system is used as input, its fitting effect on most algorithms is better than that of the single-level indicator system, and their scores are relatively high. The performance values of the two-level indicator system are 0.7749, 0.0389, 0.85, 1.0, and 0.4365 in terms of R^2 and 0.0085, 0.0366, 0.01, 0.0, and 0.0214 for MSE, which shows an improvement compared with the single-level indicator system's performance values (0.8924, 0.0301, 0.84, 1.0, and 0.1263 for R^2 and 0.0041, 0.0369, 0.01, 0.0, and 0.0333 for MSE). As mainly reflected in Lasso, SVR, Dtree, and especially MLP, it has made excellent progress; it only has reduced performance with Ridge, but the decline is not obvious. All these results prove that our two-level indicator system is effective for the commonly used methods.

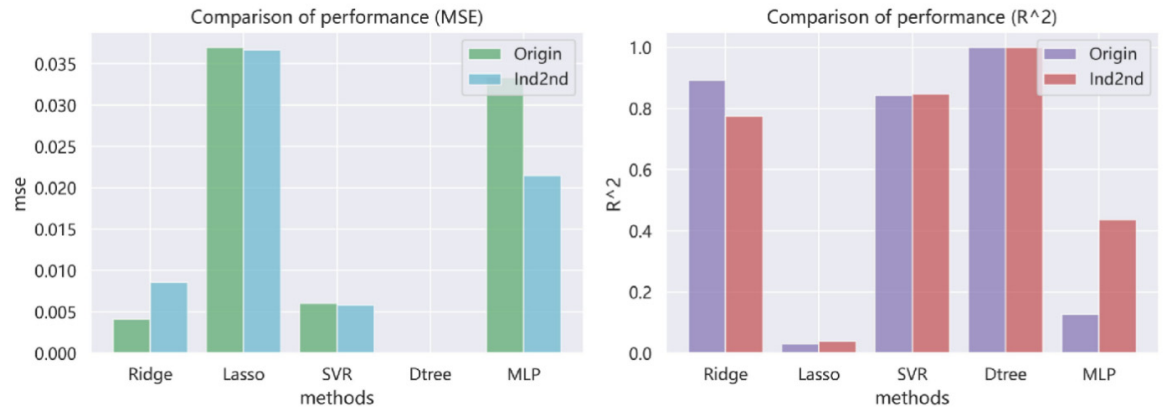

Figure 6. Fitting performance of single-level and two-level indicator systems. The origin is the single-level indicator system, Ind2nd is the two-level indicator system.

However, there is an anomaly in which the performance of Lasso and MLP appears to be significantly lower than the other methods. This is because they are treated in a special way during the evaluation. The algorithms we use are all from Python's sklearn library, where an alpha parameter exists in "linear_model.Lasso" and "neural_network.MLPRegressor". We also found that the setting of this parameter would cause a huge change in the results. Thus, for these two methods, we averaged them 100 times by modifying their alpha values in the range of 0–1; they were extremely low because of their erratic performance. Figure 7 shows how they perform on MSE and R^2 with an alpha between 0 and 1.

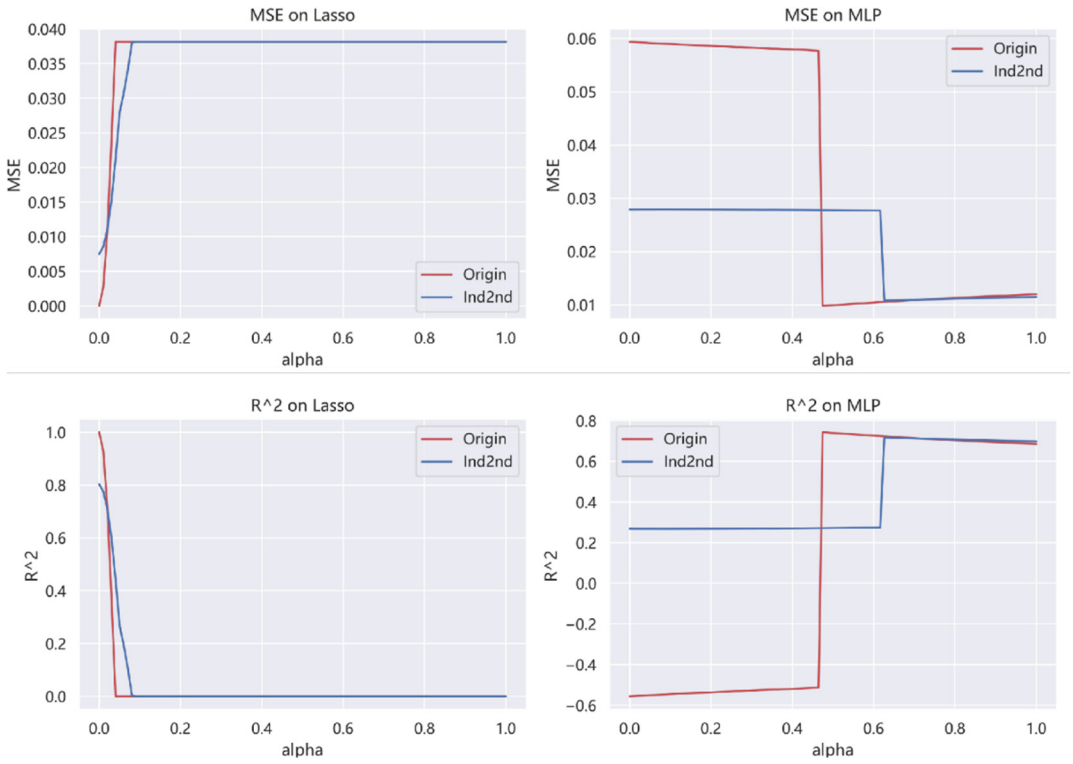

Figure 7. Performance of Lasso and MLP when alpha is between 0 and 1.

As can be seen from Figure 7, only looking at R^2, Lasso's performance attenuates rapidly when the alpha exceeds a certain threshold, whereas the change is relatively slow in the two-level indicator system, which also explains why our method can win by a narrow margin. However, the optimal performance (R^2 = 0.8 when alpha = 0) is not comparable to that of the single-level indicator system (R^2 = 1 when alpha = 0). In fact, Lasso degenerates into classical linear regression when alpha = 0, which proves that the performance of the two-level indicator system under the linear regression algorithm is also weaker than that of the single-level indicator system. On the other hand, there are also abrupt changes in performance in MLP, but our system outperforms the single-level indicator system over a long range and equates with the single-level indicator system at the top. Therefore, the latter explains why the two-level indicator system is superior to the single-level indicator system in MLP. At the same time, it can be seen that the variance of the performance curve of the second-level indicator system is the smallest. This means that our method can perform better than the single-level index system in both Lasso and MLP in most cases, even though both indicator systems are extremely

picky about the alpha when running on Lasso (it only works if the alpha is within a specified small range). This is due to the lack of expression ability of Lasso itself. However, no matter what the value is of the MLP, we can still get a relatively satisfactory result, for two reasons: First, the performance of the MLP itself is better, and it can ensure stable operation no matter what kind of data it faces. Second, from the perspective of fitting, our two-level indicator system synthesizes the contribution of all first-level indicator systems to the final result, resulting in better results obtained by the model.

In summary, by comparing the fitting effects of the constructed two-level indicator system and the original single-level indicator system on different algorithms, our idea can be seen as effective: (1) The weight network of information security risk assessment indicators established by the random forest algorithm can effectively depict the dependency relationship between indicators, which can help us understand the whole indicator system more comprehensively and clearly understand the status and role of indicators in the whole system. (2) Spectral clustering was carried out based on the weight network, and the roles and dependencies of different indicators in the system were further divided. The single-level system is divided into the two-level system ($SubGraph_1 = [\mathrm{A14}, \mathrm{A15}, \mathrm{A16}]$, $SubGraph_2 = [\mathrm{A1}, \mathrm{A2}, \mathrm{A3}, \mathrm{A6}], SubGraph_3 = [\mathrm{A4}, \mathrm{A7}, \mathrm{A9}, \mathrm{A13}], SubGraph_4 = [\mathrm{A5}, \mathrm{A8}, \mathrm{A10}, \mathrm{A11}, \mathrm{A12}]$) and retained their dependency relationship. (3) Finally, PageRank was used to lock important indicators, and corresponding core indicators were found to be the security of the operating system (A15), the authenticity of the information content (A5), the controllability of the information content (A6), and the public security consciousness (A9). Finally, differences among algorithm models are compared, showing that the constructed indicator system has better stability and accuracy than the original indicator system when applied to common algorithms. This indicator system can be applied to most scenarios. In fact, we can attempt to explain this two-level indicator system. We only need to put the key indicators searched by PageRank into the subgraph for observation. Taking $SubGraph_1$ as an example, it includes firewall reliability (A14), operating system security (A15), and vulnerability threat repair rate (A16). Among these three indicators, the operating system security is definitely affected by the other two factors. Given that firewall and vulnerability threats are the most direct factors affecting the security of an operating system, this finding is consistent with the logic. Meanwhile, as can be seen from Figure 5, this indicator is also directly affected by network resource connection (A2) outside the subgraph, and indirectly affected by communication network construction (A1) and urban cloud platform construction (A3), which are important external environments to ensure the security of the operating system. The same can be said for $SubGraph_2$. When the construction of communication network platforms and connected resources tends to be stable and secure, it can provide us with continuous controllable information. The same goes for $SubGraph_3$ and $SubGraph_4$.

5. Conclusions

With the deepening of urban informatization, the number of indicators for information security risk assessment is growing rapidly, and existing models become insufficient in assisting decision-makers as they cannot effectively analyze the internal dependence of indicators and their levels. To solve this problem, we propose a graph-based two-level risk assessment indicator system construction method. This method uses the random forest algorithm to extract the dependency network of indicators from the dataset and obtains the two-level indicator system through network clustering and PageRank. Finally, the single-level indicator system and the constructed two-level indicator system are applied and compared using typical algorithms. The results show that the two-level indicator system

outperforms the single-level indicator system.

Specifically, we collected a total of 16 indicators from 25 cities in China and constructed the indicators' dependency network. Then, the subsequent network clustering and PageRank analysis showed that the indicator system can be made up of four subnetworks: the security of the operating system (A15), the authenticity of the information content (A5), the controllability of the information content (A6), and the public security consciousness (A9). Based on the above results, we suggest strengthening the construction and security inspection of infrastructure and network facilities to ensure that data are running on secure devices. Moreover, backup and recovery technology of information in circulation and operation should be strengthened to ensure that information does not lose its original meaning or cannot be tampered with. Finally, we offer the following recommendations: (1) enhancing security education and information security consciousness of people; (2) encouraging scientific production and application of information technology; (3) providing a good external environment for the deep integration of science and technology.

6. Future work

There are many promising future directions for using deep learning and machine learning methods in the construction of smart cities, such as smart city construction, smart management, smart education, privacy, and security. We know that training models can provide accurate results when similar feature sets and distribution models form training and test data. Researchers should also focus on the integration of semantic technologies in smart city applications to enable smart devices to better interact with users. For information security reasons, federated learning (FL), differential privacy (DP), and secure multi-party computing (SMC) are novel approaches, as they enable encrypted sharing of data between different departments. Combining federal learning with smart city applications can provide privacy and protection of sensitive information, enhancing the security of information exchange between various city departments.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This study was supported by the National Social Science Fund of China (Grant No. 18BTQ055), the Postgraduate Scientific Research Innovation Project of Hunan Province (CX20210544) and the Social Science Foundation of Hunan Province (18YBA398).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. J. Bokolo, Data driven approaches for smart city planning and design: a case scenario on urban data management, *Digital Policy Regul. Governance*, **25** (2023), 351–367. https://doi.org/10.1108/dprg-03-2022-0023
2. A. A. Semlambo, D. M. Mfoi, Y. Sangula, Information systems security threats and vulnerabilities: A case of the Institute of Accountancy Arusha (IAA), *J. Comput. Commun.*, **10** (2022), 29–43. https://doi.org/10.4236/jcc.2022.1011003.
3. J. Andress, *Foundations of Information Security: A Straightforward Introduction*, No Starch Press, San Francisco, 2019.
4. A. Chiniah, F. Ghannoo, A multi-theory model to evaluate new factors influencing information security compliance, *Int. J. Secur. Networks*, **18** (2023), 19–29. https://doi.org/10.1504/IJSN.2023.129949
5. T. Finne, A conceptual framework for information security management, *Comput. Secur.*, **17** (1998), 303–307.
6. A. Herzog, N. Shahmehri, Towards secure e-services: Risk analysis of a home automation service, in *6th Nordic Workshop on Secure IT-Systems*, (2001), 18–26.
7. H. Zhu, S. Liu, Y. Qu, X. Han, W. He, Y. Cao, A new risk assessment method based on belief rule base and fault tree analysis, in *Proceedings of the Institution of Mechanical Engineers*, **236** (2022), 420–438. https://doi.org/10.1177/1748006X211011457
8. X. Xu, F. Yu, W. Pedrycz, X. Du, Multi-source fuzzy comprehensive evaluation, *Appl. Soft Comput.*, **135** (2023), 110042. https://doi.org/https://doi.org/10.1016/j.asoc.2023.110042
9. H. Liu, Z. Zhang, Z. Sun, A fuzzy comprehensive evaluation model for smart city application, *Int. J. Innovative Comput. Appl.*, **11** (2020), 96–102. https://doi.org/10.1504/ijica.2020.107120
10. O. T. Arogundade, A. Abayomi-Alli, S. Misra, An ontology-based security risk management model for information systems, *Arab. J. Sci. Eng.*, **45** (2020), 6183–6198. https://doi.org/10.1007/s13369-020-04524-4
11. H. Taherdoost, A review on risk management in information systems: Risk policy, control and fraud detection, *Electronics*, **10** (2021), 3065. https://doi.org/10.3390/electronics10243065
12. A. Tantawy, S. Abdelwahed, A. Erradi, K. Shaban, Model-based risk assessment for cyber physical systems security, *Comput. Secur.*, **96** (2020), 101864. https://doi.org/10.1016/j.cose.2020.101864
13. K. Tam, K. Jones, MaCRA: A model-based framework for maritime cyber-risk assessment, *WMU J. Marit. Aff.*, **18** (2019), 129–163. https://doi.org/10.1007/s13437-019-00162-2
14. Y. Tang, M. Elhoseny, Computer network security evaluation simulation model based on neural network, *J. Intell. Fuzzy Syst.*, **37** (2019), 3197–3204. https://doi.org/10.3233/jifs-179121
15. W. Cai, H. Yao, Research on information security risk assessment method based on fuzzy rule set, *Wireless Commun. Mobile Comput.*, **2021** (2021). https://doi.org/10.1155/2021/9663520
16. K. Dixit, U. Singh, B. Pandya, Comparative framework for information security risk assessment model, in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022*, (2022). http://doi.org/10.2139/ssrn.4121814
17. R. Wirtz, M. Heisel, Model-based risk analysis and evaluation using CORAS and CVSS, in *International Conference on Evaluation of Novel Approaches to Software Engineering*, **1172** (2020), 108–134. https://doi.org/10.1007/978-3-030-40223-5_6

18. A. S. Alfakeeh, A. Almalawi, F. J. Alsolami, Y. B. Abushark, A. I. Khan, A. A. S. Bahaddad, et al., Hesitant fuzzy-sets based decision-making model for security risk assessment, *Comput. Mater. Continua*, **70** (2022), 2297–2317. https://doi.org/10.32604/cmc.2022.020146
19. R. Kaur, D. Gabrijelčič, T. Klobučar, Artificial intelligence for cybersecurity: Literature review and future research directions, *Inform. Fusion*, **97** (2023), 101804. https://doi.org/10.1016/j.inffus.2023.101804
20. J. Song, H. Xu, Safety risk evaluation of tourism management system based on PSO-BP neural network, *Wireless Commun. Mobile Comput.*, **2023** (2023). https://doi.org/10.1155/2023/2968129
21. Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, X. Liang, An improved random forest based on the classification accuracy and correlation measurement of decision trees, *Expert Syst. Appl.*, **237** (2024), 121549. https://doi.org/10.1016/j.eswa.2023.121549
22. G. Zhong, C. Pun, Self-taught multi-view spectral clustering, *Pattern Recognit.*, **138** (2023), 109349. https://doi.org/10.1016/j.patcog.2023.109349
23. T. Chapuis-Chkaiban, Z. Toffano, B. Valiron, On new PageRank computation methods using quantum computing, *Quantum Inf. Process.*, **22** (2023), 138. https://doi.org/10.1007/s11128-023-03856-y

AIMS Press