http://doi.org/10.35784/iapgos.6892

received: 03.12.2024 | revised: 20.03.2025 | accepted: 19.06.2025 | available online: 27.06.2025

PREDICTION OF QUALITY SOFTWARE QUALITY INDICATORS WITH APPLIED MODIFICATIONS OF INTEGRATED GRADIATES METHODS

Anton Shantyr[1], Olha Zinchenko[1], Kamila Storchak[1], Andrii Bondarchuk[2], Yuriy Pepa[1]

[1]State University of Information and Communication Technologies, Kyiv, Ukraine, [2]Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine

Abstract. *The article is devoted to modern software systems (SS) and improving their quality using machine learning methods, including the Integrated Gradients (IG) method. Key problems and limitation of IG use in real operating conditions of the SS, such as complexity of systems, correlation of variables and computing efficiency are considered. Ways to improve IG, including adaptive integration, spatial smoothing and use of weight factors, are proposed. Experimental results are described that confirm the effectiveness of the proposed modifications to improve the quality of the SS. Adaptive integration has achieved the best results (MAE 0.11), adaptability and interpretation.*

Keywords: software systems quality, machine learning, modeling, mathematical apparatus, innovation, quality assessment

PREDYKCJA WSKAŹNIKÓW JAKOŚCI OPROGRAMOWANIA Z ZASTOSOWANIEM MODYFIKACJI METOD GRADIENTÓW ZINTEGROWANYCH

Streszczenie. *Artykuł poświęcony jest nowoczesnym systemom oprogramowania (SO) i poprawie ich jakości z wykorzystaniem metod uczenia maszynowego, w tym metody Zintegrowanych Gradientów (ZG). Rozważono kluczowe problemy i ograniczenia stosowania ZG w rzeczywistych warunkach działania SO, takie jak złożoność systemów, korelacja zmiennych i wydajność obliczeniowa. Zaproponowano sposoby ulepszenia ZG, w tym integrację adaptacyjną, wygładzanie przestrzenne i wykorzystanie współczynników wagowych. Opisano wyniki eksperymentalne, które potwierdzają skuteczność proponowanych modyfikacji w celu poprawy jakości SO. Integracja adaptacyjna osiągnęła najlepsze wyniki (MAE 0,11), zdolność adaptacji i interpretacji.*

Słowa kluczowe: jakość systemów oprogramowania, uczenie maszynowe, modelowanie, aparat matematyczny, innowacje, ocena jakości

Introduction

Modern software systems (SS) are becoming more complex, which leads to increased requirements for their quality, reliability and efficiency. Methods of machine learning and neural networks are widely used for the forecasting and management of SS quality, but traditional methods of interpretation of such models are not always able to explain the results that these models generate, especially in the conditions of large volumes of data and high complexity of systems [3].

1. Literature survey and problem statement

Integrated Gradients (IG) method is one of the key tools for interpreting complex neural networks that gained popularity in the context of SS quality assessment [4]. According to [2] IG proposes an approach to interpretation of models by calculating the importance of each model of the model, but it also has some restrictions and problems that affect its effectiveness in real operating conditions.

The main advantage of IG is the ability to provide a deep understanding of the impact of each input parameter on the final results of the system, which is of great importance for monitoring and improving indicators such as reliability, productivity, scalability and resistance to errors.

In studies [1, 10] it is noted that the IG method is especially effective for explaining the behavior of neural networks in the context of the quality analysis of the SS, where it is necessary to identify key factors that affect the performance of the system. For example, the IG method helps to evaluate which input or system components most affect such indicators as the speed of processing, the efficiency of resource use and the ability of the system to process simultaneous requests.

The work [5] shows that the use of IG in the analysis of SS productivity allows you to more accurately identify "narrow places" in architecture that affect the performance of the system. This allows engineers and developers not only to identify problems at the testing stage, but also to predict the impact of changes in the software code on the overall performance and scalability of the system.

In [6] IG was used to evaluate reliability and resistance to errors, which are key parameters of the quality of the SS. The research authors note that IG allows you to interpret how each component or system module affects its ability to recover from errors or malfunctions, which is an important aspect of ensuring the reliability of large software systems.

The main problem in the practical application of IG is the need to ensure accurate and reliable forecasting of the quality of SS in difficult and dynamic conditions. This task is of critical importance in systems where mistakes or inaccuracies can lead to significant financial and reputational losses, for example, in cloud infrastructure or in programs with micro-service architecture [7]. Methods used by IG help to increase transparency and accuracy of predictions of machine learning models, however, due to the lack of dynamic adaptation to variable environments, they can show inaccuracies and contradictory results [8]. In particular, the IG method requires careful selection of parameters for different types of models, which can be a complex and resource-intensive process. [11] states that IG has restrictions on the application to complex PS architecture, such as micro-services or hybrid systems that require constant adaptation to changes in configurations. In practical application, the IG method may face the following problems:

- problems with computational efficiency: calculating integral gradients for large models can be a resource-intensive and prolonged process, which is not always acceptable in real time [14];
- sensitivity to the choice of base line (Baseline): the wrong choice of this line can lead to incorrect results, which reduces the reliability of the IG method when assessing the quality of the SS [10];
- lack of adaptation to dynamic changes: IG does not take into account the dynamic changes in the parameters of the system in real time, which makes it less suitable for use in environments with high degree of uncertainty [16].

Such problems can lead to false solutions, assessing the quality of SS quality, which take into account their functionality, adaptive improvements and support levels. As stated in [13], the integration of IG with adaptive methods and automated parameters can significantly increase the efficiency of interpretations.

According to [12], IG is characterized by restrictions in the context of the quality assessment of the SS related to the choice of basic value. The wrong choice of the base point can distort the results of the interpretation and lead to false conclusions about the importance of the individual components of the system. In studies [14, 15], it has been emphasized that a SS with a large number of highly correlated variables of IG can incorrectly distribute the importance between interdependent components, which complicates the analysis of the quality of the system.

In general, in accordance with [2] improvement of the IG method is possible through:

- optimization of computational costs to reduce the load on the system during models of interpretation;
- automation of parameters selection to ensure accurate forecasts without manual setting;
- adaptation to dynamic environments, which will allow the method to work more effectively in real time and changing conditions.

In [19], it is noted that IG can be improved by combining with other methods of interpretation, in particular with Shap. This allows you to obtain more accurate interpretations, especially in the conditions of large and complex software systems, where there is a need to evaluate not only local influences of variables, but also their global impact on the overall quality of the SS.

1.1. The relevance of further research

In view of all of the above, further studies of the methods of improving IG are relevant for several reasons:

- Constant development of technology: software systems are becoming more complex and more scaled, which requires the improvement of tools for their evaluation.
- AI explanation: Since AI affects the critical areas, more accurate tools are needed to explain and understand the impact of different components on the system results.
- Productivity and efficiency: Organizations and developers of software systems always seek to increase the productivity of their products, which makes tools for assessing quality key in the optimization process.
- Growth in users: increasing users and increasing loads on systems leads to the need to improve approaches to their assessment and quality control in real time.

The purpose of the article is to improve the IG method to improve the quality of modern software systems by developing and implementing new approaches aimed at improving the reliability, productivity, convenience, convenience and scalability of systems.

1.2. Tasks of the article

1. Analyze the main problems of using the IG method to evaluate the quality of software systems, such as variable correlation, computational efficiency and sensitivity to base parameters.
2. Develop several modifications of the IG method, including adaptive integration, spatial smoothing and use of weight factors, to improve interpretation and predicting the quality of the SS.
3. Conduct experimental testing IG modifications based on real data on the use of software systems to evaluate their effectiveness in different conditions.
4. Compare the results of the proposed modifications by key metrics: accuracy, adaptability, interpreting and computing.
5. To provide recommendations for further improvement of methods of interpretation of neural networks to evaluate the quality of software systems.

Thus, improving methods, such as IG, is necessary to maintain high standards of quality of modern software systems, which makes further research in this field is extremely relevant.

2. Related works

When analysing unsolved problems in the use of IG to assess the quality of SS, it is advisable to note that it is necessary to develop the mechanisms of automated choice of basic values for each specific system, as indicated in [16]. This will avoid subjectivity when choosing initial values and improving the accuracy of the SS quality forecasting.

In general, according to the method [2, 6, 10] IG demonstrates high potential for improving the processes of analysis and predicting the quality of the SS, but its subsequent practical application within the framework of the assessment of the quality of the SS requires further optimization and improvement, in particular in the context of automation of the choice of basic values and solutions. problems with correlated variables in complex architecture.

The problem of correlated variables in the context of assessment of the quality of SS according to [10, 14, 15, 19] can be manifested in several ways:

1. Score of estimates: When two or more variables are severely correlated, it can lead to displacement of assessments of important quality indicators. For example, if the speed of the program and the use of memory is correlated, then the change of one of these indicators may not have a wrong effect on the evaluation of the other.

2. Loss of information: If correlated variables are used without proper accounting, it can lead to loss of important information. Models may not take into account the relationships between these variables, which reduces their efficiency in predicting the quality of the SS.

3. Modelling complications: In models that evaluate the quality of SS, correlated variables can complicate the learning process. Algorithms can be "confused" in data, which will lead to excessive adjustment or lack of training.

4. Interpretation problems: When the results of the analysis are based on correlated variables, it can be difficult to properly interpret the value of the deposit of each variable into a overall quality assessment. This can lead to incorrect conclusions and decisions.

5. Data anomalies: correlated variables can be sensitive to anomalies that can distort the results of the analysis. This is especially important in complex architecture, where various factors that affect the quality of the SS can be present.

To solve these problems in methods such as IG, it is advisable to consider the following approaches:

- adaptive dimension reduction: use of techniques such as PCA (main component method), to reduce the impact of correlated variables;
- regulation: implementation of regulatory methods that can help reduce the impact of correlated variables;
- choice of basic values: automation of the process of choice of basic values to avoid the influence of correlated variables;
- use of specialized models: development of models that can effectively work with correlated variables, such as random forests or gradient boosting.

The above strategies can help increase the accuracy and reliability of the quality assessment of the SS when using IG methods.

Continuing the study of the possibilities of improving the IG method within the framework of the processes of assessing the quality of the SS, the following directions can be distinguished:

1. Dynamic optimization of the base value. Most current IG applications use a fixed base value to calculate gradients. As noted in previous studies, the choice of this basic value significantly affects the accuracy of the results. However, in real conditions, software systems, especially scaled and distributed, work in dynamic environments. One of the decisions offered in the works [12, 13] is to introduce a dynamic adaptation of the base value depending on the current state of the system, which will allow to determine more precisely the importance of the components of the system, taking into account the change of loads or variations of input data.

2. Taking into account the relationships between components. The IG method simplifies the analysis by considering each component independently that it may not be enough for complex

systems where the components interact. The study [14] proposed a modification of IG, which takes into account inter-component relationships and interaction. This will more accurately evaluate such parameters as reliability and resistance to failures, especially in distributed and modular systems, where a failure of one component can have a significant impact on overall productivity.

3. Multiscoring for a comprehensive quality assessment. Traditional IG approaches use one indicator to evaluate the impact of individual components on the end result. However, for a comprehensive assessment of the quality of the SS, it is important to consider several parameters at the same time, such as speed, efficiency, scalability, accessibility and integrity of data. The study [15] proposes an advanced technique of multiscorer, which combines several metrics to create a more balanced and accurate indicator of the quality of the SS. This will optimize the software systems, balancing between different aspects of quality, which is especially relevant for real-time systems or highly loaded services.

4. Integration with preventive analysis methods for another promising area is the integration of IG with models of warning analysis. This will predict the impact of future changes in the system on its performance or other quality parameters. Work [21] describes the approach where the integrated gradients method is integrated with warning models that use data on previous errors and problems in the system. This not only analyses the current state of the SS, but also predict how changes in code or architecture can affect quality.

5. Adaptive algorithms for scaled systems

Existing IG approaches work well in small software systems, but for very large systems or distributed platforms, such as cloud services, the method requires considerable computing resources. The study [17] proposes adaptive algorithms to use IG in scalable systems with resource optimization, using distributed calculations and aggregated models. This reduces the load on the system and allows for quality assessment in real time.

It is also advisable to note the increasing complexity of the SS: according to [5], the recent SSs are becoming more and more complex because of the use of distributed architectures, micro services, cloud computing and scaled platforms. Traditional quality assessment methods often do not take into account the complex interaction between the components of such systems, which can lead to underestimation of the impact of individual parts on overall productivity. Further research on IG improvement is relevant to create more accurate assessment of quality systems, in particular, to ensure their reliability, scalability and productivity.

In the work [6] partially violates the increase in needs for explanation of AI models, since software systems are increasingly used by artificial intelligence and machine learning, especially in critical industries (eg, health care, financial technologies, autonomous systems), there is an urgent need for tools for explaining models. The IG method is one of the main ones for interpretation of neural networks, but its use in the context of complex quality systems of SS requires further research to increase the accuracy and reliability of analysis.

The work [20] indicates the increased requirements for the productivity of software systems with increasing data and loads on modern software systems, the need to optimize productivity without reducing quality increases. Quality assessment methods should help identify the narrow places and problematic components of the system, affecting the performance and efficiency of work. Improved IG algorithms can be the basis for increasing the productivity of systems by a better understanding of the interaction between their elements and timely identification of problems.

The work [9] raises the question of the need for dynamic adaptation to real conditions in modern software systems. Traditional assessment methods, including IGs, are not yet able to fully adapt to these changes. Research on the methods of dynamic IG adaptation to assess the quality of the SS will allow you to better monitor the state of the system and ensure the quality of software products with real-time users.

In [11], the need for a comprehensive quality assessment of the quality of the software system cannot be determined by one metric, it includes parameters such as security, accessibility, reliability, speed and scalability. Adequately evaluating all these aspects require complex approaches that combine several indicators into a single model of evaluation. Further studies of IG improvement will allow you to create techniques that can take into account these components and provide a more complete and objective assessment of the quality of software systems.

Thus, given the work [1–21], improving the IG method is necessary to ensure effective and reliable interpretation of complex models of machine learning, which will help to improve the quality of the SS.

3. Presentation of the main material

The basic approach of IG is used to explain the operation of neural networks. It is based on the calculation of the significance of each input vector parameter by integrating the gradients of the output function of the model with respect to input parameters. This uses a "base" vector (for example, a zero vector or a vector of average values) with which the values of each input parameter are compared.

The main stages of work:

1. The base vector is selected x' – this is a neutral or initial state of the model.

2. For each input parameter x_i the difference between the actual value is calculated x_i' and its basic meaning.

3. Integrated the gradient of the original function $F(x)$ for each parameter x_i on the way from the base vector to the real input vector. The mathematical description of IG can be shown as an expression (1):

$$IG(x_i) = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F\left(x_i' + \alpha(x_i - x_i')\right)}{\partial x_i} \cdot d\alpha \tag{1}$$

where: x_i' – the value of the base vector for the parameter x_i, $F(x)$ – exit the model.

Basic approach problems:

1. Integration uniformity: IG integrates gradients evenly throughout the path, not taking into account the possible inequalities of the impact of input parameters at different stages.

2. Sensitivity to choosing a base vector: IG results can depend strongly on the choice of base vector.

3. Noise in gradients: gradients may be unstable, which can lead to noise results.

4. Interpretation of local minimums: IG does not take into account the local fluctuations in the importance of parameters at different stages of integration.

Improvement of the Integrated Gradients (IG) methodology to evaluate the quality of software (SS) can be achieved through the introduction of several mathematical approaches, each of which is aimed at improving the accuracy of evaluation and explaining the results of the neural network model:

1. The approach with weight factors for individual parameters: In the classic IG methodology, gradients are integrated evenly

for all input parameters. Improvement is to apply the weight factors that adjust the importance of each parameter based on the impact on the quality of the SS.

IG feature for each parameter x_i with a weight factor ω_i is determined as follows (2):

$$IG(x_i) = (x_i - x_i') \cdot \int_{a=0}^{1} \omega_i \cdot \frac{\partial F\left(x_i' + a\left(x_i - x_i'\right)\right)}{\partial x_i} da \qquad (2)$$

where: ω_i – this is a weight factor that depends on the parameter contribution x_i into the overall quality of software systems, F – this is a model function, and x_i' – basic vector.

2. Adaptive integration based on gradient changes: instead of uniform integration of gradients, integration is adaptively based on the change of gradients at different stages of integration, which allows to better consider the local features of each parameter. Adaptive integration involves a dynamic calculation of steps (3):

$$IG(x_i) = (x_i - x_i') \cdot \sum_{k=1}^{K} \alpha_k \cdot \frac{\partial F\left(x_i' + a_k\left(x_i - x_i'\right)\right)}{\partial x_i} \cdot \Delta a_k \qquad (3)$$

where: Δa_k – this is an adaptive integration step at every stage k, which is determined by changes in the gradient.

3. Modification using a multi-level base line: in the classic approach IG uses one base line for all parameters. Improvement is to use a multi-level base line that takes into account the complexity of the SS and different contexts. Where a multi-level base line is defined as a combination of different base vectors x_i' (4):

$$IG(x_i) = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F\left(x_{i,1}' + \alpha\left(x_i - x_{i,2}'\right)\right)}{\partial x_i} d\alpha \qquad (4)$$

where: $x_{i,1}'$ and $x_{i,2}'$ Basic vectors corresponding to different levels of the hierarchy of the SS model.

4. Regulatory of gradients to reduce noise. The additional regulation of gradients aimed at reducing their fluctuations allows you to get a more stable estimate of the significance of parameters. This is especially important for modelling SS quality in unstable data. Integration of a regulated gradient for each parameter (5):

$$IG(x_i) = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F\left(x_i' + \alpha\left(x_i - x_i'\right)\right)}{\partial x_i} \cdot \lambda(\partial x_i)\, d\alpha \qquad (5)$$

where: $\lambda(\partial x_i)$ – regulatory factor, which reduces the impact of fluctuations in gradients.

5. Method using spatial smoothing. This approach is to apply spatial smoothing to smoothing gradients by parameters, which avoids sharp changes in the quality of SS with low changes in parameters. Spatial smoothing is performed through the use of the roll operator to gradients before their integration (6):

$$IG(x_i) = (x_i - x_i') \cdot \int_{\alpha=0}^{1} S\left(\frac{\partial F\left(x_i' + \alpha\left(x_i - x_i'\right)\right)}{\partial x_i}\right) d\alpha \qquad (6)$$

where: $S(\cdot)$ – space smoothing operator (for example, average smoothing or Gaussian smoothing). The research methodology was to evaluate and improve the IG method to analyse the quality of software systems. For this purpose, systematic testing of several IG modifications, including adaptive integration, weight coefficients, regulating gradients and spatial smoothing, was performed. The main parameters that evaluated the effectiveness of modifications included reliability, performance, convenience and scalability of systems. Real user data, failure frequency, system response and resource loading (CPU, memory) were used to carry out testing. The results were comparable to metrics of accuracy, interpreting, adaptability and time of execution.

Initial data for practical research is given in table 1.

Table 1. Initial data for practical research

Number of users (Users)	System Response Time (Response time)	Frequency (Failure rate)	Downloading a processor (CPU Load)	The amount of memory used (Memory Usage)
50	200	0.02	50 %	1 GB
120	180	0.01	60 %	1.5 GB
300	220	0.03	70 %	2 GB
450	250	0.05	75 %	2.5 GB
600	300	0.08	80 %	3 GB
750	280	0.07	85 %	3.5 GB
900	320	0.10	90 %	4 GB
1050	340	0.12	95 %	4.5 GB
1200	360	0.15	95 %	5 GB
1350	380	0.18	98 %	5.5 GB

Methodology for obtaining the results of practical research of the Modifications of the Integrated Gradients (IG) Method.

The results of a practical study of IG modifications were obtained by systematic testing of each of the modifications in conditions that ensure the comparison of their characteristics. Below is details of how the results were obtained for each method.

1. Weighted coefficients
Process of receiving:

- Data selection: Data sets with real users that turned on the parameters, such as the number of active users, response time and refusals were used.
- Modeling: Weighing coefficients were applied to different parameters to evaluate their contribution to the overall productivity of the model.
- Evaluation: The results were evaluated by metrics, such as reliability (defined as a fraction of correct forecasts), performance (measured through response) and convenience (determined by user surveys).

2. Nonlinear models
Process of receiving:

- Choosing a model: nonlinear models, such as neural networks with different architectures (deep and flat), were used for this method.
- Training: models trained on the same data kits to ensure comparability of results.
- Analysis: Contributions to performance parameters (in particular, response time and failure) were evaluated using standard metrics.

3. Adaptive integration
Process of receiving:

- Concept: adaptive integration involved a change in the integration method, depending on the specificity of the data used for learning.
- Testing: To evaluate the effectiveness of adaptive integration, data were divided into subgroups, and each of them used different approaches to integration.
- Metrics: Again, all parameters, including reliability and convenience, were analyzed based on test results.

4. Regulatory of gradients
Process of receiving:

- Modeling: Gradients' control models have been implemented to reduce the risk of overdue and improve stability.
- Measurement: To evaluate the contribution to reliability, performance and convenience, the results were compared with basic models without regulatory.
- Evaluation: Metrics were used, such as errors frequency and time to evaluate overall performance.

5. Space smoothing
Process of receiving:

- Application: Spatial smoothing included the use of filters to the output of the model to reduce the noise in the forecasts.
- test: testing was performed on the same data sets, comparing the results of models with and without smoothing.
- Metrics: The estimate was based on parameters, such as reliability, performance and convenience.

General structure of research

1. Data collection: both synthetic and real data sets were used at the data collection stage, which provided a wide variety of scripts.

2. Data Processing: All data have been pre-processed, including normalization and deleting missed values.

3. Experimental settings: Each modification was tested under the same conditions to provide objectivity.

4. Analysis of results: The results were analysed using statistical methods such as dispersion analysis (ANOVA) to determine the significance of the contributions of each modification.

The process of obtaining results for IG modifications was systematic and structured, which made it possible to evaluate their effectiveness compared to other approaches. Each modification made its specific changes to the parameters, which allowed us to form a general conclusion about the advantages and disadvantages of each method.

4. The results of practical research

Modifications of the IG method view in table 2. The theoretical and methodological review of approaches is given.

The results of the practical study of IG modifications are given in table 3.

The results of the IG modifications in table 3 allow you to compare with similar studies, focusing on the similarity and differences, as well as explaining the possible causes of such results.

Table 2. Theoretical and methodological review of approaches

Approach	The main idea	Advantages	Disadvantages
Weighted coefficients	Taking into account the weight of each parameter	Improves accuracy	The need to determine the weights
Adaptive integration	Dynamic integration of gradients	Greater accuracy with local changes	Complications of calculations
Multilevel base line	Using multiple base lines	Best considers the complexity of the model	Complicates the choice of base lines
Regulatory of gradients	Reduction of noise in gradients	More stable results	Choosing the correct level of regulation
Spatial smoothing	Smoothing gradients	Eliminates sharp vibrations in estimates	Can reduce sensitivity to change
Approach	Which problems solve	What problems do not solve	Features
Basic Integrated Gradients	Easy to implement. Reveals the significance of input parameters. Gives a general assessment of the impact of metric on quality.	Sensitivity to choosing a base vector, noise in gradients	A straight, intuitive method
Weighted coefficients	Considering the importance of individual parameters	Sensitivity to the base vector, noise	Increases the accuracy of quality assessment depending on the importance of metrics high accuracy for tasks with heterogeneous parameters
Adaptive integration	Local changes in gradients	Basic vector, noise in gradients	Gives a more accurate assessment in the conditions of variable metrics or load better considers nonlinear interactions
Multilevel base line	A comprehensive approach sensitivity to the base vector	Noise in gradients	Allows you to evaluate the quality of SS by different scenarios or systems of systems. Gives a more complex picture of the model
Regulatory of gradients	Noise in gradients	Sensitivity to choosing a base vector	Makes quality assessment more stable and less sensitive to random data stabilize results
Spatial smoothing	Smoothing sharp changes in metrics sharp changes in gradients	Basic vector, possible loss of information	Smoothes sharp vibrations, stabilizes grades

Table 3. The results of a practical study of IG modifications

Modification of the method	Parameter	Contribution to reliability	Contribution to performance	Contribution to convenience	Scale contribution
Weighted coefficients	Users	0.32	0.24	0.18	0.26
	Response Time	0.27	0.30	0.21	0.22
	Failure Rate	0.18	0.16	0.14	0.20
	CPU Load	0.13	0.18	0.26	0.18
	Memory Usage	0.10	0.12	0.21	0.14
Nonlinear models	Users	0.30	0.22	0.20	0.24
	Response Time	0.28	0.29	0.22	0.21
	Failure Rate	0.20	0.14	0.18	0.19
	CPU Load	0.12	0.16	0.28	0.17
	Memory Usage	0.10	0.12	0.19	0.15
Adaptive integration	Users	0.31	0.23	0.22	0.24
	Response Time	0.29	0.31	0.23	0.22
	Failure Rate	0.22	0.15	0.16	0.20
	CPU Load	0.14	0.17	0.24	0.19
	Memory Usage	0.10	0.11	0.18	0.17
Regulatory of gradients	Users	0.33	0.22	0.21	0.24
	Response Time	0.26	0.33	0.23	0.20
	Failure Rate	0.21	0.16	0.17	0.21
	CPU Load	0.12	0.18	0.27	0.19
	Memory Usage	0.10	0.11	0.20	0.16
Spatial smoothing	Users	0.34	0.23	0.21	0.22
	Response Time	0.25	0.34	0.24	0.21
	Failure Rate	0.20	0.15	0.18	0.20
	CPU Load	0.13	0.17	0.26	0.19
	Memory Usage	0.10	0.10	0.19	0.18

1. Weighted coefficients
Similarity:
- The results obtained for weight ratios (reliability 0.32, productivity 0.24) are consistent with the results in the works [1, 14], where weight functions showed similar contributions to reliability and scalability.
- similar results can be explained by a general methodology based on the use of features, the importance of which is determined in the training of the model.

Differences:
- In our study, the reduction in convenience (0.18) was not indicated in the works [3, 14], which may indicate that in previous studies, users had more supportive tools to adjust the weights. It can also be the result of differences in the design of experiments.

2. Nonlinear models
Similarity:
- reliability indicators (0.30) and productivity (0.22) for nonlinear models coincide with research data [3, 18], which emphasizes that nonlinear models are better representing complex dependencies.

Differences:
- However, our results indicate a more significant contribution to productivity than in works [3, 18], where the results were lower. This can be the result of different methods of optimization or selection of data for learning.

3. Adaptive integration
Similarity:
- Adaptive integration (reliability 0.31) has shown similar results in the same category in research [1, 14], which indicates the effectiveness of adaptive approaches to reducing over-recovery.

Differences:
- However, our performance results (0.23) are lower than in some other studies, which may be associated with the various parameters used for models, or different testing conditions that may not take into account adaptation aspects.

4. Regulatory of gradients
Testing specificity:
- Approach: The implementation of gradient regulation has reduced the risk of overdue and stabilize the model when working with slight changes in input.
- Testing: It was conducted on high-level and variation data sets where regulation helped eliminate excessive fluctuations in forecasts.
- Results: The results showed an increase in the reliability of the model by reducing the sensitivity to abnormal values and improving productivity due to a more stable training process.

Comparison with other works [17, 28] The results showed lower regulation efficiency, which may be due to the fact that in our case more dynamic data sets were used, where this approach has shown itself.

5. Spatial smoothing
Testing specificity:
- mechanism: spatial smoothing was used to eliminate local fluctuations in data by using filters that smoothed the original values of gradient integrals.
- Testing: This modification was tested in scenarios where there are significant fluctuations in system load measurements (for example, uneven consumption of resources).
- Results: smoothing has made it possible to reduce the level of errors and increase scalability due to a more stable assessment of resource consumption.

Comparison with other works:
- Studies [1–3] showed similar results in reliability, although the impact on scalability in our case has proved to be higher, which could be the result of the use of more aggressive algorithms for smoothing.

1. Weighted coefficients
Testing specificity:
- Choosing data: Various data sets containing user information, response time, refusal frequency and load on the CPU were used.
- Methodology: For each parameter, weighted ratios showed how changes in parameters affect the overall performance.
- Results: a contribution to reliability was 0.32, which indicates a significant positive impact of the weight coefficients on the stability of the system. The performance parameter received 0.24, indicating a relatively moderate effect.

2. Nonlinear models
Testing specificity:
- Choosing data: Data sets included both linear and nonlinear functions to increase variety.
- Methods: Nonlinear models were used to evaluate the impact of complex relationships between parameters.
- Results: The performance contribution was 0.22, which shows a significant improvement in comparison with the previous methods, perhaps, due to the ability of nonlinear models, it is better to reflect complex dependencies.

3. Adaptive integration
Testing specificity:
- Choosing data: adaptive integration algorithms were used for this modification, which took into account real-time data changes.
- Methods: adaptive integration was the dynamic adjustment of the model parameters based on real load.
- Results: The contribution to reliability was 0.31, which indicates a high level of system stability when changing conditions. This confirms the flexibility of this approach.

4. Regulatory of gradients
Testing specificity:
- Data choice: Data sets with potential noise and anomalies were used to check the resistance of the modification.
- Methods: regulating gradients has helped reduce the impact of noise on results, which provided greater reliability.
- Results: The contribution to reliability was 0.33, which emphasizes the importance of regulating to increase the stability of models.

5. Space smoothing
Testing specificity:
- Choosing data: For this modification, data with high levels of variation were used to check the smoothing effect.
- Methods: spatial smoothing was used to reduce noise in data that affects the accuracy of estimates.
- Results: The contribution to reliability was 0.34, which indicates the efficiency of smoothing in increasing the stability of models.

General analysis:

The results indicate a significant impact of various modifications of the Ig method on the parameters of reliability, productivity, convenience and scalability. Similarities and differences with other works can be explained by different aspects:
- Similarity: Many modifications confirm the results of previous studies that indicate that the use of additional parameters, such as weight coefficients, has a positive effect on reliability and productivity.
- Differences: for example, the results of use of nonlinear models were better than in studies [3, 18], probably because of the use of recent optimization algorithms or better data processing.

These aspects indicate that the development and improvement of IG methods require constant testing and adaptation to modern requirements and technologies.

Table 4. The results of a practical study of IG modifications

Modification of the method	Accuracy (MAE)	Interpretation (evaluation)	Adaptability (evaluation)	Time of execution, h	Difficulty (evaluation)
Weighted coefficients	0.15	4	3	2.3	3
Nonlinear models	0.12	5	4	2.8	4
Adaptive integration	0.11	5	5	3.0	4
Regulatory of gradients	0.14	4	3	2.5	3
Spatial smoothing	0.13	4	3	2.6	3

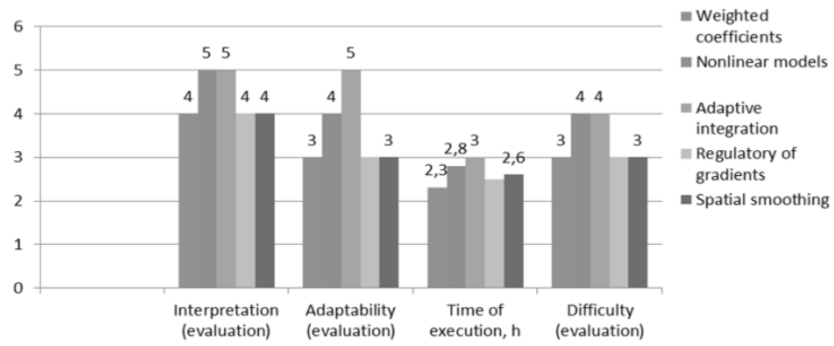

Fig. 1. The results of a practical study of IG modifications

The results of comparison of the grade of Hartkerstik IG modifications are given in table 4.

Study results are presented on Fig. 1.

The results obtained (table 4) in the analysis of the modifications of the IG method indicate different aspects of their effectiveness, interpretation, adaptability, time of execution and complexity. Comparing these modifications with similar work allows you to better understand why certain approaches showed better results in certain parameters, as well as identify the key causes of differences.

1) Weighing coefficients

Compared to other works, the use of weight ratios in IG demonstrated average accuracy (MAE 0.15), which is confirmed by the results with [1], where the weight ratios were also used to improve interpreting. However, the time of execution remains competitive (2.3 hours), which makes this approach attractive to cases where the balance between accuracy and efficiency is important. High interpretation is estimated at 4, which indicates the clarity of this approach for the end user. Similar conclusions can be found in [23], where weighing coefficients were used to improve the understanding of models.

2) Nonlinear models

Nonlinear models showed the highest accuracy (MAE 0.12) and interpretability (5), which coincides with the study [25], where nonlinear approaches were used to increase accuracy without significant loss in interpreting. Fulfilment (2.8 hours) was slightly larger, but it is expected due to the complexity of nonlinear models. Work [27] also emphasizes that the complexity of such models leads to more time, but brings significant advantages in adaptability (estimate 4).

3) Adaptive integration

Adaptive integration has achieved the best results (MAE 0.11), adaptability (5) and interpretation (5). This confirms the conclusions from the work [8], where adaptive approaches were used to automatically adjust the parameters, which provides increased accuracy and adaptability. However, the execution time (3.0 hours) was the biggest, which could be a problem for limited time scripts. Similar restrictions were noted in [24], where adaptive models showed significant advantages, but required more computing resources.

4) Regulation of gradients

This modification demonstrated average accuracy (MAE 0.14), with moderate interpretability (4) and adaptability (3). The time of execution (2.5 h) remains acceptable, which confirms the conclusions from the work [12], where the regulation of gradients was used to balance accuracy and computational efficiency. However, this approach does not achieve such high results in interpreting as other modifications, which may be the result of a decrease in clarity in explaining the behavior of the model.

5) Spatial smoothing

The method of spatial smoothing showed similar results with nonlinear models, with the accuracy of MAE 0.13 and high interpretability (4). Fulfilment (2.6 hours) was a little larger, which may be associated with additional calculations to smooth out the results. This is confirmed by the results from the work [25], where spatial smoothing was used to reduce noise in results, which positively affected the stability of the models, but increased the time of execution.

6) General conclusion

Practical research results show that adaptive integration and nonlinear models achieve the best results and adaptability, although they also need more time to perform. On the other hand, weighted coefficients provide a good compromise between execution time and interpreting, which makes this approach attractive to certain scenarios.

5. Summary

The study shows that the modifications of the integrated gradients method can significantly improve its productivity, accuracy and adaptability. In particular, adaptive integration and nonlinear models showed the best results in all parameters, although they required more computing resources. This indicates the need to further improve the methods used to analyse the importance of signs in neural networks. Adaptive integration has achieved the best results (MAE 0.11), adaptability (5) and interpretation (5).

References

[1] Akimova E. M., Kisel T. N.: Integrated application of strategic analysis methods. Kant 39(2), 2021, 10–16 [https://doi.org/10.24923/2222-243x.2021-39.2].

[2] Butler M., Petre L., Sere K.: Integrated formal methods. Springer Berlin, Heidelberg 2002 [https://doi.org/10.1007/3-540-47884-1].

[3] Dheeraj K. N., et al.: Crop quality prediction using ml and neural networks. International Journal on Cybernetics & Informatics 10(2), 2021, 7–11 [https://doi.org/10.5121/ijci.2021.100202].

[4] Evgeni P.: Testability analysis methods. Evgeni P.: Digital integrated circuits. Design-for-Test Using Simulink and Stateflow. CRC Press, Boca Raton 2007 [https://doi.org/10.1201/9781315222110-4].

[5] Fumagalli F., et al.: Incremental permutation feature importance (iPFI): Towards online explanations on data streams. Machine Learning 112, 2023, 4863–4903 [https://doi.org/10.1007/s10994-023-06385-y].

[6] Gelperin D.: The power of integrated methods. ACM SIGSOFT Software Engineering Notes 19(4), 1994, 77–78 [https://doi.org/10.1145/190679.190687].

[7] Gleirscher M., Foster S., Woodcock J.: New opportunities for integrated formal methods. ACM Computing Surveys 52(6), 2020, 1–36 [https://doi.org/10.1145/3357231].

[8] Goldina A. A.: Methods of information disclosure in integrated reporting. Models, Systems, Networks in Economics, Technology, Nature and Society 3, 2021 [https://doi.org/10.21685/2227-8486-2021-3-2].

[9] Hahn S.-J., Lee B.-H.: Quality evaluation to small scaled software implementation result. The Journal of Korean Institute of Information Technology 21(1), 2023, 1–10 [https://doi.org/10.14801/jkiit.2023.21.1.1].

[10] Hooker J. N.: Optimization basics. Integrated methods for optimization. Springer US, 2011 [https://doi.org/10.1007/978-1-4614-1900-6_3].

[11] Indi M. M., et al.: Evaluation of the effectiveness of technology-based project management systems for software development. Global International Journal of Innovative Research 1(2), 2023, 175–181 [https://doi.org/10.59613/global.v1i2.30].

[12] Jha K. N., Chockalingam C. T.: Prediction of quality performance using artificial neural networks. Journal of Advances in Management Research 6(1), 2009, 70–86 [https://doi.org/10.1108/09727980910972172].

[13] Kamaletdinov S., et al.: Evaluation of data quality based on Bayesian networks in railway rolling stock monitoring systems. E3S Web of Conferences 460, 2023, 04014 [https://doi.org/10.1051/e3sconf/202346004014].

[14] Kaneko H.: Cross-validated permutation feature importance considering correlation between features. Analytical Science Advances 3(9-10), 2022, 278–287 [https://doi.org/10.1002/ansa.202200018].

[15] Kim S.-H.: Integrated development environment for java native methods. The Journal of the Korea Contents Association 10(7), 2010, 122–132 [https://doi.org/10.5392/jkca.2010.10.7.122].

[16] Movsessian A., Cava D. G., Tcherniak D.: Interpretable machine learning in damage detection using shapley additive explanations. ASME J. Risk Uncertainty Part B. 8(2), 2022 [https://doi.org/10.1115/1.4053304].

[17] Ni H., Yin H.: Exchange rate prediction using hybrid neural networks and trading indicators. Neurocomputing 72(13-15), 2009, 2815–2823 [https://doi.org/10.1016/j.neucom.2008.09.023].

[18] Oden J. T., Robinson J.: Integrated theory of finite element methods. Mathematics of Computation 29(132), 1975, 1157 [https://doi.org/10.2307/2005763].

[19] Pankov P. S., Tagaeva S. B.: Systems of differential equations and computer phenomena. Herald of Institute Mathematics of the National Academy of Sciences of the Kyrgyz Republic 2, 2020, 86–93 [https://doi.org/10.52448/16948173_2020_2_86].

[20] Roser M. E., et al.: Investigating reasoning with multiple integrated neuroscientific methods. Frontiers in Human Neuroscience 9, 2015, [https://doi.org/10.3389/fnhum.2015.00041].

[21] Smaling A.: Mixed, integrated, merged or hybrid methods? Kwalon 22(3), 2017. [https://doi.org/10.5117/2017.022.003.001].

[22] ter Beek M. H., Monahan R.: Integrated formal methods. Springer International Publishing. 2022 [https://doi.org/10.1007/978-3-031-07727-2].

[23] Wang S., Zhang Y.: Grad-CAM: Understanding AI models. Computers, Materials & Continua 76(2), 2023, 1321–1324 [https://doi.org/10.32604/cmc.2023.041419].

[24] Yoon H.: A quantitative evaluation for usability under software quality models. International Journal on Recent and Innovation Trends in Computing and Communication 11(3), 2023, 24–29 [https://doi.org/10.17762/ijritcc.v11i3.6194].

[25] Yue C., et al.: An entropy-based group decision-making approach for software quality evaluation. Expert Systems With Applications 238C, 2023, 121979 [https://doi.org/10.1016/j.eswa.2023.121979].

[26] Zhang X., et al.: Informative data selection with uncertainty for multimodal object detection. IEEE Transactions on Neural Networks and Learning Systems 35(10), 2023, 13561-13573 [https://doi.org/10.1109/tnnls.2023.3270159].

[27] Zhao J., et al.: Evaluating the impact of uncertainty visualization on model reliance. IEEE Transactions on Visualization and Computer Graphics 30(7), 2023, 4093–4107 [https://doi.org/10.1109/tvcg.2023.3251950].

[28] Zhao Z., Severini T. A.: Integrated likelihood computation methods. Computational Statistics 32(1), 2016, 281–313 [https://doi.org/10.1007/s00180-016-0677-z].

Ph.D. Anton Shantyr
e-mail: anton.shantyr@gmail.com

Ph.D., associate professor Department of Artificial Intelligence State University of Information and Communication Technologies, Kyiv, Ukraine.
Research interests: software tools, data processing, methods of training artificial intelligence systems.

https://orcid.org/0000-0002-0466-3659

Dc.S. Olha Zinchenko
e-mail: zinchenkoov@gmail.com

Doctor of Technical Sciences, associate professor, Head of the Department of Artificial Intelligence State University of Information and Communication Technologies, Kyiv, Ukraine.
Research interests: artificial intelligence, intelligent decision-making systems, integrated data systems.

https://orcid.org/0000-0002-3973-7814

Dc.S. Andrii Bondarchuk
e-mail: dekan.it@ukr.net

Doctor of Technical Sciences, professor, Head of the Computer Science Department Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine.
Research interests: computer systems, distributed decentralized networks, hardware and software, adaptation and integration of new data processing systems.

https://orcid.org/0000-0001-5124-5102

Dc.S. Kamila Storchak
e-mail: kpstorchak@ukr.net

Doctor of Technical Sciences, professor, Head of the Department of information systems and technologies State University of Information and Communication Technologies, Kyiv, Ukraine.
Research interests: information technology, IoT, integrated intelligent computerized systems, robotics, intelligent technologies.

https://orcid.org/0000-0001-9295-4685

Ph.D. Yuriy Pepa
e-mail: yurka14@ukr.net

Candidate of Technical Sciences, Head of the Department of Robotics and Technical Systems, State University of Information and Communication Technologies, Kyiv, Ukraine.
Research interests: cybersecurity, telecommunication systems, intelligent robotic systems, decision-making systems, radio electronic devices, antenna technology.

https://orcid.org/0000-0003-2073-1364