BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 73(5), 2025, Article number: e154286
DOI: 10.24425/bpasts.2025.154286

ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE

Enhancing transparency and user-interactivity in sentiment analysis design through X-OODM

Abqa JAVED[1], **Muhammad SHOAIB**[1], **Abdul JALEEL**[2], **Salman JAN**[3]*, **Ahmed ALKHYYAT**[4], and **Ali SAMAD**[5]

[1] Department of Computer Science, University of Engineering and Technology, 54890 Lahore, Pakistan
[2] Department of Computer Science (RCET GRW), University of Engineering and Technology, 52250 Lahore, Pakistan
[3] Faculty of Computer Studies, Arab Open University, A'Ali, 732, Bahrain
[4] College of Technical Engineering, The Islamic University, Najaf 54001, Iraq
[5] Department of Data Sciences, Faculty of Computing, The Islamia University of Bahawalpur, Pakistan

Abstract. Explainability is a significant factor in the realm of web-based applications. It provides a robust system for understanding and interpreting the internal functioning of applications throughout the design process. Nonfunctional parameters are integrated into the transparent and user-interactive models presented in X-OODM. Various components are employed to generate the metrics for each parameter, which then serve to develop the overall model metric. However, X-OODM used different scenarios of web-based applications as a case study to assess design quality metrics. In this study, we used domain diagrams from C0 to C10 as design models, improved with various sentiment analysis use cases, to assess the applicability of X-OODM and related metrics. Each domain diagram presents a distinct functionality that is evaluated under the user-interactive model and the transparent model of X-OODM. The user-interactive model uses transferability, informativeness, and accessibility, whereas the transparent model includes simulatability, decomposability, and algorithmic transparency. These parameters are further classified into several components, all of which contribute to the explainable model. A multiple linear regression is used to assess the explainable metric for each class domain model. The robust user-interactive and transparent model metrics determine the statistical significance for the design of web-based applications, specifically in sentiment analysis. This work can be extended to implement all the X-OODM models for the evaluation of web-based applications.

Keywords: transparency; user-interactivity; sentiment analysis and measures; explainability; object-oriented design methodology.

1. INTRODUCTION

The object-oriented design (OOD) methodology emphasizes modeling software systems as collections of interacting objects, each with its own state and behavior [1]. Various object-oriented methodologies exist at the design level to devise and develop a system [2]; however, the existing methodologies are not specifically designed for web-based applications [3–5]. OODM is designed for web-based applications by defining classes and objects [6] based on the waterfall model with object-oriented principles. These models are not evaluated at the design level to ensure the complete working of each component. Security metrics are implemented in [7] with the methodology guaranteeing that the application is secure and protected at the design phase. These metrics assist in identifying the potential areas of refinement, which lead towards more effective and efficient design.

Although metrics help developers, they do not facilitate the users' getting insights into the working models [8]. Explainability implies the ability to describe the inner workings of a model in a human-understandable language [9]. The focus of explainable models is to provide interpretability, fairness, and transparency in a computing system [10]. The OODM faces significant challenges regarding explainability, making it difficult to understand and justify complicated designs [11]. One challenge is the complicated structure of object-oriented design to understand the links and interactions between objects [12]. While abstraction and encapsulation principles promote modularity, they may add to complexity in design. Furthermore, as systems become more complex, the enormous number of interconnected objects [13] and their activities could prevent straightforward understanding. Addressing these issues is critical to increasing transparency in the object-oriented design process. Improved explainability can facilitate developer collaboration and assist in debugging, maintenance, and ethical considerations, resulting in more understandable, trustworthy, and ethically acceptable software systems [14]. Integrating explainability [15] into the design level of OODM can promote transparency, collaboration, and early detection of design issues, thereby enhancing the quality, transparency, and fairness of the system. When design decisions are clearly explained [16], users can confidently interact with the system and obtain the details of each feature.

To address the limitations of OODM, the explainability component is introduced in explainable object-oriented design methodology (X-OODM) [17, 18]. It incorporates explainabil-

*e-mail: salman.jan@aou.org.bh

Manuscript submitted 2024-07-15, revised 2025-04-13, initially accepted for publication 2025-05-04, published in August 2025.

ity in the analysis and design phases of web-based applications. This expansion is achieved by incorporating explainable components into the X-OODM framework, as indicated in Fig. 1. X-OODM facilitates the early detection of potential design flaws or shortcomings, enabling timely corrective actions.

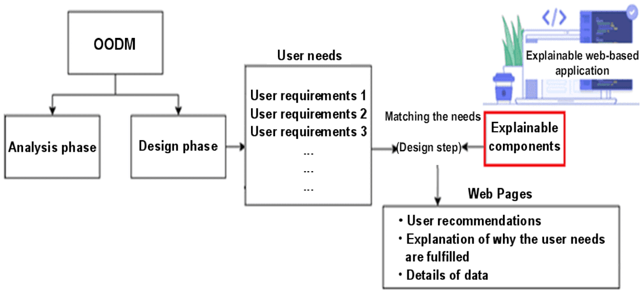

Fig. 1. Explainable design phase of OODM web-based applications

This study contributes to the X-OODM framework by enhancing its explainability components and aligning them with sentiment analysis. Some of the key improvements include the inclusion of more advanced metrics on the user-interactive and transparent models, which are transferability, informativeness, accessibility, simulatability, decomposability, and algorithmic transparency metrics. These metrics are validated further using multi-domain diagrams (C0 to C10 – extended for sentiment analysis), to enhance the explainability at the design level. We applied a robust multiple linear regression model to quantify the effects of explainability parameters to provide better statistical rigor. Comparative evaluations bring forth measurable gains in design transparency and user trust, thereby establishing that the extended framework meets all the demands of modern web-based applications.

The improved metrics of the explainable model help in detecting potential shortcomings or biases early in the development process of web-based sentiment analysis design, allowing for timely adjustments to enhance transparency and the user interactivity. Moreover, quantifying the X-OODM ensures that the design process is rigorous, transparent, and aligned with developers and users' expectations.

We summarized the following contributions in the domain of OODM for web-based applications:

- Explainable components introduced in X-OODM are enhanced for the transparency of the web-based applications at the user level.
- Design quality metrics are improved for the explainable components at an early stage of design to ensure the quality of the explainable components.
- We validated these improved metrics with multidomain sentiment analysis web-based applications, where results proved that the metrics are accurately quantified to provide proficient explainability to the users.
- This explainable model is viable to implement in every web-based application. Moreover, these metrics are helpful to compare other explainable designs to provide better quality.

The paper is organized as follows: the next section discusses proposed work in materials and methods, Section 3 presents results and discussion, and Section 4 concludes the paper.

2. MATERIALS AND METHODS

In the proposed work, the explainable framework is modified by introducing the features of sentiment analysis such as feedback module, at the design level for web-based applications. The details of explainability features or parameters are given in Section 2.1, which is further used in the quantification of the proposed metrics.

2.1. Explainability parameters

Explainability in the web-based applications are used as a quality indicator as it provides detailed information in the analysis and the design phase of the application. Accurate explanation of the components enhances the trust of the users and improves the interpretability [18–20].

The explainability parameters presented in X-OODM [17] and [18], given in Table 1, cover all aspects of the explain-

Table 1
Explainability parameters used at design level OODM

Parameters	Description
Transferability	Transferability enhances the ability of the system to communicate information. This provides features that allow users to provide domain-specific data to improve the refined model of sentiment analysis and promote its efficient generalization over several domains [21].
Informativeness	Informativeness will ensure that the interface clearly, relevantly, and efficiently shares information with the users. The aim is to enable the users to understand, trust, and believe in how the system works and its functionality [22, 23].
Accessibility	The goal is to present the data in an uncomplicated way so that even users who do not have knowledge about data analysis can easily understand the simple explanation behind their data. It uses simple words to make one understand the output from an application without using complicated terms, which might be difficult for stakeholders to comprehend [9].
Simulatability	Simulatability in a transparent model is critical for allowing users to comprehend and replicate the decision-making process of the model. This boosts users' trust in the model predictions by allowing them to check the accuracy and fairness of the outcome [24].
Decomposability	Decomposability plays a vital role in assisting people in understanding how a model works since it divides it into components. By studying each individual feature, detail, or rule, users may discover how it influences the predictions of the model. As a result, the model must be part of a system made up of explicit components that are simple for humans to understand and manage [25].
Algorithmic transparency	Algorithmic transparency is incredibly important in an explainable model, so that people can trust its predictions. It focuses on explaining to the user how the algorithm behaves and makes decisions [26, 27].

ability for web-based applications. In this work, we used user-interactive and transparent models with their parameters and components. Myriad studies [28–30] focus on the explainability features to ensure privacy and security, along with legal compliance, to get insight into internal activities. In the user-interactive model, different parameters are used, including transferability, informativeness, and accessibility. Whereas, under the transparent model, simulatability, decomposability, and algorithmic transparency are measured as explainable features.

2.2. Relationship of models, parameters, and components in an explainable domain

To achieve an improved and significant explainability in web-based applications, X-OODM models have several parameters and related components which contributes to achieve the explainability at design level. These models combine the input of each parameter and related components, as depicted in Fig. 2, to collectively measure the explainability of a system under observation.

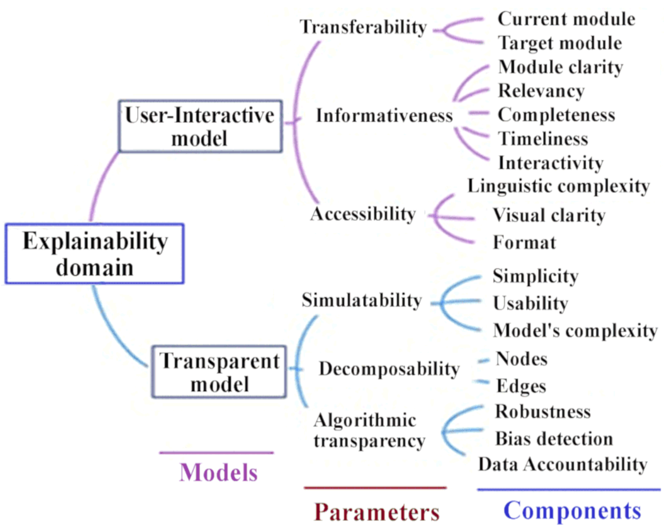

Fig. 2. Explainable domain, models, and association of explainability

Each model parameter has further different components that are used to quantify the model. The applications are used by several users, and the parameters that have been established guarantee that the implementation of the various components inside the application can be explained. These components include clarity, relevancy, timeliness, interactivity, simplicity, usability, robustness, and accountability. The corresponding components obtain information from a range of factors and transfer them to the relevant model. This makes it possible to implement an explainable model, which improves system efficiency and offers the user an interactive, transparent, safe, and interpretable system. These parameters can be used to evaluate the explainability of the system. Moreover, additional parameters can be provided based on the user's requirements. Furthermore, a description of each component along with the factors defined for the parameters is provided in Table 2.

Table 2
Components and related factors for explainability

Components	Factors	Description
Current to target module transition	Save backend and frontend logs, current modules, and target modules.	Effective transition of data from the current to the target modules.
Module clarity	Clarify functionality.	Each module of the application is clearly understandable by the users.
Relevancy	Validate data. Ensure relevancy.	The application provides the related information.
Completeness	Limit information.	Information provided in all aspects.
Timeliness	Ensure timeliness.	Provide up-to-date information.
Interactivity	Enhance interactivity.	Each component interacts with the users.
Linguistic contextuality	Ensure linguistic.	Provide an understandable language of application.
Visual clarity	Enhance visuality.	Effectively provides a clear and detailed visualization of each module.
Format accessibility	Data consistency.	Data within the application is easily readable.
Simplicity	Evaluation simplicity.	Each module provides basic and simple details.
Usability	Enhance usability.	The application is usable for everyone.
Complexity of the model	Measure complexity.	Provide a clear interface for the users.
Nodes	Identify nodes.	Number of connected nodes within the application.
Edges	Identify edges.	Connected edges with each module.
Robustness	Evaluate robustness.	Ensure a robust system.
Bias detection	Error handling. Measure bias.	The application is biased, free data.
Data accountability	Data accountability. Backup recovery.	Data accountability is implemented within each module.

Figure 3 depicts the explainability domain model (class diagram 0) with explainable components tailored for web-based sentiment analysis applications. This diagram serves as a foundation for evaluating the impact of explainability metrics and outlines the calculation and assessment methodology of this work. Overall, we used class diagrams (C0 to C10), each designed with a different set of explainability components to address specific explainability criteria in various user applications, including interpretability and user understanding.

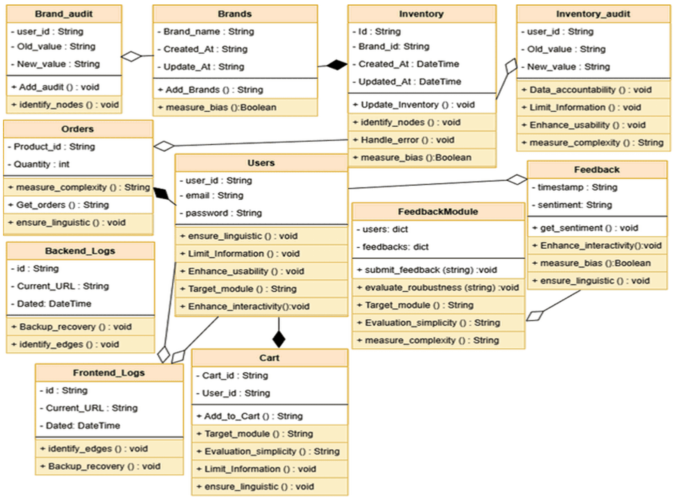

Fig. 3. Domain model with explainable components (class diagram 0)

2.3. Quantifying the explainable user interactive model (XUIM)

Quantification of the user-interactive model to introduce explainability uses different parameters and their components. It helps to enhance the accessibility, transferability, and informativeness of the application for end-users. Each metric is considered as a linear model and has an equal impact on overall explainability; however, the impact can be adjusted with weighted factors.

2.3.1. Transferability

Equation (1) presents the transferability metrics by measuring the current module and the target module. The average impact of transferability is achieved in (2).

$$\text{Target metrics} = \varphi_i\,, \qquad \text{Current metrics} = \rho_i\,,$$

$$\text{Transferability metrics} = \sum_{i=1}^{n}\left(\frac{(\varphi_i - \rho_i)}{\varphi_i} * 100\right), \tag{1}$$

$$\text{Average transferability metrics} = \frac{\sum_{i=1}^{n}\left(\frac{(\varphi_i - \rho_i)}{\varphi_i} * 100\right)}{n}. \tag{2}$$

2.3.2. Informativeness

The informativeness metric is defined using clarity, relevance, completeness, timeliness, and interactivity in (3), and the average ratio is defined in (4).

$$\text{Clarity metrics} = C_i\,, \qquad \text{Relevance metrics} = \pi_i\,,$$

$$\text{Completeness metrics} = \theta_i\,, \qquad \text{Timeliness metrics} = T_i\,,$$

$$\text{Interactivity metrics} = N_i\,,$$

$$\text{Informativeness metrics} = \sum_{i=1}^{n}(C_i + \pi_i + \theta_i + T_i + N_i)\,, \tag{3}$$

$$\text{Average informativeness metrics} = \frac{\sum_{i=1}^{n}(C_i + \pi_i + \theta_i + T_i + N_i)}{n}. \tag{4}$$

2.3.3. Accessibility

Equation (5) defines comprehensive accessibility measurements, which include the essential elements of linguistic relevancy, visual clarity, and format usability. The average ratio of accessibility metrics is calculated using (6).

$$\text{Linguistic contextuality metrics} = L\emptyset_i\,,$$

$$\text{Visual clarity metrics} = Vc_i\,,$$

$$\text{Format accessibility metrics} = \circ\mathrm{F}\nabla_i\,,$$

$$\text{Accessibility metrics} = \sum_{i=1}^{n}(L\emptyset_i + Vc_i + \circ\mathrm{F}\nabla_i)\,, \tag{5}$$

$$\text{Average accessibility metrics} = \frac{\sum_{i=1}^{n}(L\emptyset_i + Vc_i + \circ\mathrm{F}\nabla_i)}{n}. \tag{6}$$

Various parameters and components contribute equally to the quantification of the user interactive model. All the parameter metrics, including (1), (3), and (5), are merged to measure the overall impact of the model.

$$\mathrm{XUIM} = \left[\sum_{i=1}^{n}\left(\frac{(\varphi_i - \rho_i)}{\varphi_i} * 100\right)\right] + \left[\sum_{i=1}^{n}({}^{\circ}C_i + \pi_i + \theta_i + \beth_i + \aleph_i)\right]$$
$$+ \left[\sum_{i=1}^{n}(L\emptyset_i + Vc_i + {}^{\circ}\mathrm{F}\nabla_i)\right]. \tag{7}$$

The XUIM metric shows the overall and average impact of the model to introduce explainability in the web-based application defined in (8) and (9).

$$\text{Overall XUIM Metrics} = \sum_{i=1}^{n}\mathrm{XUIM}_i\,, \tag{8}$$

$$\text{Average XUIM Metrics} = \sum_{i=1}^{n}\frac{\mathrm{XUIM}_i}{n}. \tag{9}$$

2.4. Quantifying the explainable transparent model (XUIM)

Metrics like simplicity, robustness, and bias assessment are used to quantify the transparent model for a web-based application, which includes simulatability, decomposability, and algorithmic transparency. By providing a numerical transparency score, these measurements guarantee that the system is transparent, equitable, and accountable for the users.

2.4.1. Simulatability

In the calculation of the simulatability of the transparent model, the parameters of the model, including simplicity, usability, and working complexity, are introduced as presented in (10).

The measure of the average ratio of simulatability is calculated in (11).

$$\text{Simplicity metrics} = SS_i\,, \qquad \text{Usability metrics} = US_i\,,$$

$$\text{Model complexity metrics} = MC_i\,,$$

$$\text{Simulatability metrics} = \sum_{i=1}^{n}\left(SS_i + US_i + MC_i\right), \tag{10}$$

$$\text{Average simulatability metrics} = \frac{\sum_{i=1}^{n}\left(SS_i + US_i + MC_i\right)}{n}. \tag{11}$$

2.4.2. Decomposability

Cyclometric metrics are measured in the decomposability metric using the nodes and edges of the model. Decomposability and average metric are defined in (12) and (13).

$$\text{Edges metrics} = \in_i\,, \qquad \text{Nodes metrics} = \cap_i\,,$$

$$\text{Connected components} = CC_i\,,$$

$$\text{Cyclometric metrics} = \in_i - \cap_i + 2CC_i\,,$$

$$\text{Decomposability metrics} = \sum_{i=1}^{n}\left(\in_i - \cap_i + 2CC_i\right), \tag{12}$$

$$\text{Average decomposability metrics} = \frac{\sum_{i=1}^{n}\left(\in_i - \cap_i + 2CC_i\right)}{n}. \tag{13}$$

2.4.3. Algorithmic transparency

Explainable models improve usability and productivity in a broad range of applications. They enable users to make informed decisions through a deeper knowledge of the model behavior, which is accomplished through algorithmic transparency, as defined by (14). The average impact of the algorithmic transparency parameter over the transparent model is measured using (15).

$$\text{Robustness metrics} = R\mu_i\,,$$

$$\text{Bias detection metrics} = BD_i\,,$$

$$\text{Data accountability metrics} = D\forall_i\,,$$

$$\text{Algorithmic transparency} = \sum_{i=1}^{n}\left(R\mu_i + BD_i + D\forall_i\right), \tag{14}$$

$$\text{Average algorithmic transparency} = \frac{\sum_{i=1}^{n}\left(R\mu_i + BD_i + D\forall_i\right)}{n}. \tag{15}$$

All the defined parameters in the transparent model are used to measure the impact of the model on the explainability value in web-based applications. The overall design metrics are combined in (16)

$$\text{XTPM} = \left[\sum_{i=1}^{j}\left(SS_i + US_i + MC_i\right)\right] + \left[\sum_{i=1}^{j}\left(\in_i - \cap_i + 2CC_i\right)\right] + \left[\sum_{i=1}^{j}\left(R\mu_i + BD_i + D\forall_i\right)\right] \tag{16}$$

The detailed XTPM in the explainable model is quantified in (17), and the average of the model is calculated in (18)

$$\text{Overall XTPM metrics} = \sum_{i=1}^{n}\text{XTPM}_i\,, \tag{17}$$

$$\text{Average XTPM metrics} = \sum_{i=1}^{n}\frac{\text{XTPM}_i}{n}. \tag{18}$$

The metrics for each parameter of the explainable model are developed and merged into their relevant models. Explainable models, such as the user-interactive model and the transparent models, have a considerable impact on the explainability of web-based applications, as shown in (19). The average impact of these models is shown in (20).

Explainable model metrics

$$(\text{XDCUITM}) = \sum_{i=1}^{n}\left(\text{XUIM}_i + \text{XTPM}_i\right). \tag{19}$$

Average explainable model metrics

$$(\text{XDCUITM}) = \frac{\sum_{i=0}^{n}\left(\text{XUIM}_i + \text{XTPM}_i\right)}{n}. \tag{20}$$

3. RESULTS AND DISCUSSION

Class diagrams are used to assess the values of the defined metrics as defined in Table 3. This quantification is helpful to achieve the explainability level of the application for the end-users and the stakeholders. The class diagrams in the banking sector utilize various parameters across different diagrams, represented as C0 to C10 [17].

These class diagrams are evaluated using the defined Algorithm 1. The values of each parameter are measured using the defined metrics of parameters with the range of 0–1 based on their functionality provided within the application defined in Table 4. Each component equally contributes to achieving the specific parameter. Each component equally contributes to achieving the specific parameter.

In object-oriented design (OOD), modular components are designed with equal importance to collectively achieve the desired functionality. For instance, in ISO/IEC 25010, which defines software quality models, each sub-characteristic contributes equally to the overall quality attribute, such as maintainability or reliability. This standardization ensures balanced evaluation without favoring any single component [31]. This

Table 3
Explainability parameters with components in each class diagram

Class diagrams	Transferability	Informativeness	Accessibility	Simulatability	Decomposability	Algorithmic transparency
C0	Target module	Completeness, interactivity	Contextuality	Simplicity, usability	Edges, nodes, complexity	Robustness, bias detection, accountability
C1	Current module	Clarity, relevance, completeness	Visual clarity, accessibility	Simplicity, usability, complexity	Nodes, complexity	Robustness, accountability
C2	Current module	Clarity, relevance, timeliness	Visual clarity, accessibility	Usability, complexity	Edges	Robustness, bias detection, accountability
C3	Current, target module	Clarity, relevance, completeness, interactivity	Contextuality, visual clarity, accessibility	Usability, complexity	Edges, nodes, complexity	Robustness, bias detection, accountability
C4	Target module	Clarity, relevance, completeness	Contextuality, visual clarity, accessibility	Simplicity, usability, complexity	Nodes, complexity	Robustness, accountability
C5	Current module	Completeness, timeliness, interactivity	Contextuality, visual clarity	Simplicity, complexity	Edges, complexity	Bias detection, accountability
C6	Current, target module	Clarity, relevance, completeness, timeliness, interactivity	Contextuality, visual clarity,	Usability, complexity	Edges, complexity	Robustness, accountability
C7	Target module	Clarity, relevance, completeness, interactivity	Contextuality, visual clarity, accessibility	Simplicity, usability,	Edges, nodes	Bias detection, accountability
C8	Current, target module	Clarity, relevance, completeness, timeliness	Contextuality, accessibility	Simplicity, usability, complexity	Edges, nodes	Bias detection, accountability
C9	Current module	Clarity, relevance, timeliness, interactivity	Contextuality, accessibility	Usability, complexity	Nodes, complexity	Robustness, bias detection
C10	Current module	Clarity, relevance, completeness, timeliness, interactivity	Contextuality, visual clarity, accessibility	Simplicity, complexity	Edges, nodes	Robustness, bias detection, accountability

Algorihm 1. Calculate explainability from class diagrams

Input: class_diagram, parameter_weights { }
Output: explainability_values { }, overall_explainability
Step 1: Parse the class diagram to extract components associated with each parameter.
Step 2: Initialize explainability_values as an empty dictionary.
Step 3: For each parameter P in parameter_weights:
  total_weight ← 0.
  For each component C associated with parameter P in the class diagram:
    if C exists in parameter_weights [P]:
      total_weight ← total_weight + parameter_weights [P][C]
  max_weight ← Maximum possible weight for parameter P.
  explainability_values [P] ← total_weight/max_weight
Step 4: overall_explainability ← sum (explainability_values ())/len(explainability_values).
Step 5: Return explainability_values, overall_explainability.

equal contribution model serves as a baseline. As more empirical data and expert evaluations become available, weights can be adjusted dynamically as given in (21) based on the actual impact of each component according to the user application-based scenario.

Values of the components and the related parameters are defined in Fig. 4 according to the different scenarios. Various numbers of users interact with the application, and the defined parameters ensure the explainability to implement numerous factors in the application. The factors include relevancy, completeness, timeliness, language relevancy, edges, and nodes. These factors provide information to the respective component, which further transfers to each related model.

This information enables the explainable model which helps to enhance the performance of the system and provides the secure, transparent, interpretable, and interactive system to the user. The systems explainability can be checked through these parameters. Moreover, these parameters can be incorporated according to users' requirements.

Table 4
Define the explainability parameters

Class diagram	Transferability	Informativeness	Accessibility	Simulatability	Decomposability	Algorithmic transparency	Explainability
C0	0.70	0.65	0.60	0.78	0.92	0.95	0.76
C1	0.85	0.78	0.72	0.97	0.78	0.89	0.83
C2	0.83	0.73	0.70	0.85	0.62	0.95	0.78
C3	0.98	0.87	0.85	0.85	0.95	0.97	0.91
C4	0.69	0.75	0.85	0.92	0.83	0.90	0.82
C5	0.85	0.82	0.77	0.89	0.85	0.85	0.83
C6	0.93	0.93	0.75	0.85	0.87	0.87	0.83
C7	0.76	0.87	0.86	0.78	0.75	0.92	0.86
C8	0.91	0.79	0.80	0.90	0.69	0.92	0.82
C9	0.82	0.86	0.78	0.86	0.82	0.75	0.81
C10	0.88	0.97	0.85	0.82	0.75	0.95	0.87

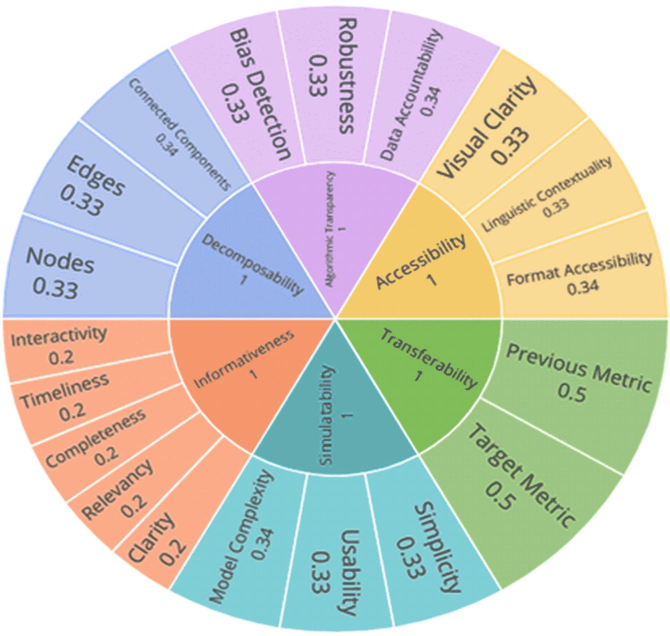

Fig. 4. Values of parameters and their related components

Detailed contextual relationships are established to develop the user interactive model and transparent model parameters for the quantification of the explainability within the web-based application. The heatmap is generated to show the correlation of independent and dependent variables in Fig. 5. The values of these parameters are clearly described as all these parameters are not dependent on each other.

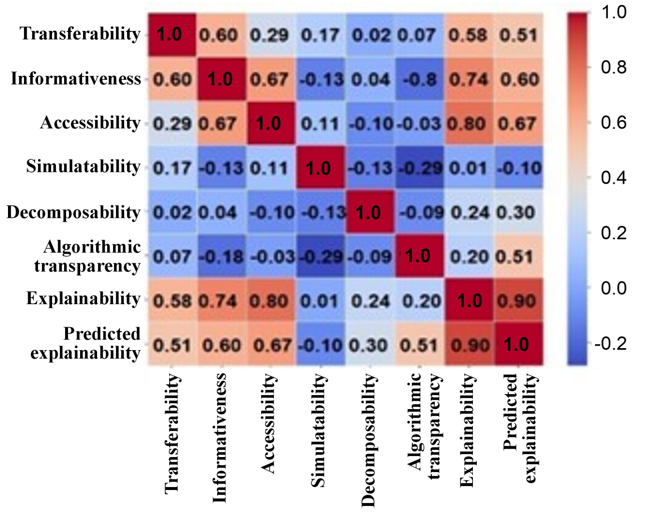

Fig. 5. Correlation matrix of parameters

3.1. Model evaluation of class diagrams

Various linear regression models are used to develop the relationship between the explainability parameters. In this work, a multivariate linear model is used for dependent and independent parameters defined in (21).

$$E_x = \varphi + \omega^1 * \sigma^1 + \omega^2 * \sigma^2 + \omega^3 * \sigma^3 + \ldots + \omega^n * \sigma^n, \tag{21}$$

where E_x is a dependent variable and the $\sigma^1, \sigma^2, \ldots, \sigma^n$ are the independent variables, $\omega^1, \omega^2, \ldots, \omega^n$ are used as a coefficient of these variables.

The coefficient defined in the equation is related to the independent variables, such as the user interactive model and the transparent model. These models further have different parameters, including transferability, informativeness, accessibility, simulatability, decomposability, and algorithmic transparency. Web-based applications are used to implement these parameters and measure explainability at the design level. Table 5 depicts the explainability value of each component using a multivariate linear model.

$$\begin{aligned} E_x &= \varphi + \omega^1 * \text{XUIMM} + \omega^2 * \text{XTPMM}, \\ E_x &= \varphi + \omega^1 * (\text{Transferability} + \text{Informativeness} \\ &\quad + \text{Accessibility}) + \omega^2 * (\text{Simulatability} + \text{Decomposability} \\ &\quad + \text{Algorithmic transparency}). \end{aligned} \tag{22}$$

The explainability is measured using (22), which defines the statistical interpretation of data and claims that the maximum explainability is achieved by implementing all the defined parameters. Table 6 provides the statistical results of the model by providing the root mean square error, R-squared, and mean absolute error.

Explainability is the dependent variable, and the other parameters are the independent variables, which are measured using the different metrics defined in (XUIMM) and (XTPMM). ω^1 and ω^2 are considered as a coefficient of the XUIMM and XTPMM, respectively. φ is the intercept, and the parameters along with the components are used to explain the variance of the explainability at the design level of web-based applications.

Table 5
Explainability values using a multivariate linear model

CD	Transferability	Informativeness	Accessibility	Simulatability	Decomposability	Algorithmic transparency	Actual explainability	Predicted explainability
C0	0.70	0.65	0.58	0.78	0.92	0.95	0.76	0.76
C1	0.85	0.78	0.72	0.97	0.78	0.89	0.83	0.81
C2	0.83	0.73	0.70	0.85	0.62	0.95	0.78	0.78
C3	0.98	0.87	0.85	0.85	0.95	0.97	0.91	0.90
C4	0.69	0.75	0.85	0.92	0.83	0.90	0.82	0.83
C5	0.85	0.82	0.77	0.89	0.85	0.85	0.83	0.82
C6	0.93	0.93	0.75	0.85	0.87	0.87	0.83	0.84
C7	0.76	0.87	0.86	0.78	0.75	0.92	0.86	0.84
C8	0.91	0.79	0.80	0.90	0.69	0.92	0.82	0.83
C9	0.82	0.86	0.78	0.86	0.82	0.75	0.81	0.80
C10	0.88	0.97	0.85	0.82	0.75	0.95	0.87	0.87

Table 6
Calculated result of the model

Model	Root mean square error	R-squared	Mean absolute error
1.	0.02776	0.11025	0.02629

The model predicted the explainability values almost same as the actual values obtained from the experts. The comparison graph shows the actual values with green color and the predicted values are represented with red color as depicted in Fig. 6.

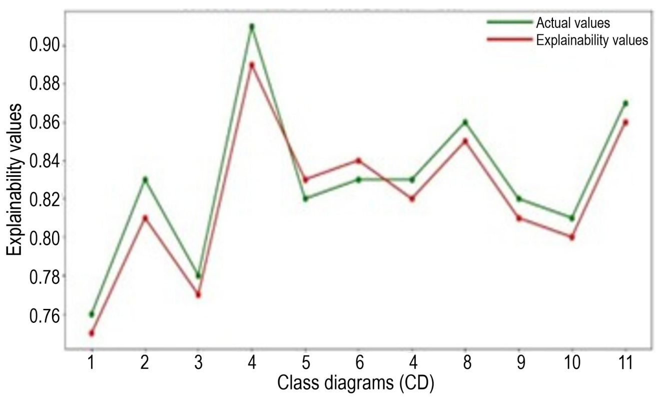

Fig. 6. Comparison of actual and predicted explainability values

3.2. Statistical analysis

To measure the validity of the proposed work for acceptance, a t-test is implemented that checks the significance of the actual explainability values and the predicted explainability values. Two hypotheses are developed to perform the t-test, and the confidence interval is measured by evaluating the two standard mean values. The output of the statistical analysis for the explainability is mentioned in Table 7. The hypothesis is designed to check the validity of the proposed work. The t-test and p-value are calculated to test the significance of the proposed model.

H0 (Null Hypothesis): There is a significant difference in explainability values.

H1 (Alternate Hypothesis): There is no significant difference in explainability values.

Table 7
T-test of explainability values

	Standard deviation	Standard error	T-test	P-value
New values	0.0390	0.0117	2.1602	0.04608
Old values	0.0410	0.0123		

The results show that there is no significant difference between the proposed explainability values and the existing values. We achieved the t-test value 2.1602, and the p-value is 0.04608, which is less than 0.05. The proposed results show that the null hypothesis is rejected, as we have the p-value within range, and the proposed model of explainability generates accurate results. Therefore, there is no significant difference in the calculated methodology and the proposed explainability methodology.

3.3. Discussion

We select transferability, informativeness, accessibility, simulatability, decomposability, and algorithmic transparency as core parameters for achieving explainability at the design level. These parameters are derived from established principles in explainable AI and software engineering methodologies, which emphasize transparency, interpretability, and user-centric understanding. By mapping these parameters to design-level artifacts, such as class diagrams and component interactions, this approach ensures that explainability is embedded from the early stages of development, rather than being limited to post-implementation evaluation. Furthermore, each parameter is achieved through sub-components, with individual weights assigned based on their involvement in the domain model. This quantitative approach not only captures the contribution of each component but also ensures flexibility and adaptability for future enhancements. The completeness of these parameters was validated through expert review and a real-world case study in web-based multi-domain sentiment analysis, where no missing factors were identified. Additionally, the alignment of these parameters with ISO 25010 Software Quality Standards further supports their sufficiency for measuring design-level explainability. Therefore, the proposed parameters provide a comprehensive and scalable framework for integrating explainability at the design level while allowing future extensions if needed.

4. CONCLUSIONS

In this paper, design quality metrics are explored for a user-interactive and transparent model proposed in web-based applications with sentiment analysis. These metrics are further analyzed using various parameters and components by ensuring their relevance to the explainability. These metrics are evaluated through domain models of web-based applications. The linear regression model is implemented on data to measure the explainability through the defined metrics. Statistical analysis is conducted to verify the significance of the proposed work, especially in the X-OODM. The results ensure the potential impact of these metrics to enhance user trust and clarity at the design level. In the future, all the other models of the X-OODM can be evaluated through different explainable AI models and machine learning models that enhance the users' trust in the design level of the applications.

REFERENCES

[1] I.J. Walker, "Requirements of an object-oriented design method," *Softw. Eng. J.*, vol. 7, no. 2, pp. 102–113, 1992, doi: 10.1049/sej.1992.0011.

[2] M. Kiewkanya, N. Jindasawat, and P. Muenchaisri, "A methodology for constructing maintainability model of object-oriented design," in *Fourth International Conference on Quality Software, QSIC, Proceedings*, 2004, pp. 206–213, doi: 10.1109/QSIC.2004.1357962.

[3] Y. Qamsane *et al.*, "A methodology to develop and implement digital twin solutions for manufacturing systems," *IEEE Access*, vol. 9, pp. 44247–44265, 2021, doi: 10.1109/ACCESS.2021.3065971.

[4] L.B. Kellner, J.V. Pilgrim, and T. Kehrer, "Round-trip migration of object-oriented data model instances," in *TTC'20: Transformation Tool Contest, Part of the Software Technologies: Applications and Foundations (STAF) federated conferences*, 2020. [Online] Available at: https://ceur-ws.org/Vol-3089/ttc20_paper1_BeurerKellner_case.pdf

[5] M. Valiante, D. Guida, M.D. Seta, and F. Bozzano, "A spatiotemporal object-oriented data model for landslides (LOOM)," *Landslides*, vol. 18, pp. 1231–1244, 2021, doi: 10.1007/s10346-020-01591-4.

[6] S. Abad, "OODM: an object-oriented design methodology for development of web applications," in *Information Modeling for Internet Applications*, IGI Global, pp. 189–229, 2003.

[7] A. Shazia, "S-OODM: A Secure Object Oriented Design Methodology", *Pak. J. Sci.*, vol. 62, no. 1, 2010, doi: 10.57041/pjs.v62i1.30.

[8] R. Ponnala, and C.R.K. Reddy, "Object Oriented Dynamic Metrics in Software Development: A Literature," *Int. J. Appl. Eng. Res.*, vol. 14, no. 22, pp. 4161–4172, 2019.

[9] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.Z. Yang, "XAI-Explainable artificial intelligence," *Sci. Robot.*, vol. 4, no. 37, p. eaay7120, 2019, doi: 10.1126/scirobotics.aay7120.

[10] R. Roscher, B. Bohn, M.F. Duarte, and J. Garcke, "Explainable Machine Learning for Scientific Insights and Discoveries," *IEEE Access*, vol. 8, pp. 42200–42216, 2020, doi: 10.1109/access.2020.2976199.

[11] M. Aniche, J. Yoder, and F. Kon, "Current challenges in practical object-oriented software design," in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, IEEE, 2019, pp. 113–116, doi: 10.1109/ICSE-NIER.2019.00037.

[12] N.L. Hsueh, J.Y. Kuo, and C. Lin, "Object-oriented design: A goal-driven and pattern-based approach", *Softw. Syst. Model.*, vol. 8, pp. 67–84, 2009, doi: 10.1007/s10270-007-0063-y.

[13] R. Ramsin, and R.F. Paige, "Process-centered review of object oriented software development methodologies," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, pp. 1–89, 2008, doi: .

[14] A. Qaisar *et al.*, "Unmasking Fake News: COVAX Reality Dataset Enrichment and BERT Mastery," in *Sustainable Data Management: Navigating Big Data, Communication Technology, and Business Digital Leadership*, Cham: Springer Nature Switzerland, vol. 2, pp. 623–632, 2025, doi: 10.1007/978-3-031-83915-3_51.

[15] R. Dwivedi *et al.,* "Explainable AI (XAI): Core Ideas, Techniques, and Solutions," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–33, 2023, doi: 10.1145/3561048.

[16] Q.V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing Design Practices for Explainable AI User Experiences," in Proc. *2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, USA, 2020, doi: 10.1145/3313831.3376590.

[17] A. Javed, M. Shoaib and A. Jaleel, "X-OODM: Explainable Object-Oriented Design Methodology," in *IEEE Access*, vol. 12, pp. 150807–150823, 2024, doi: 10.1109/ACCESS.2024.3477553.

[18] A. Javed, M. Shoaib, A. Jaleel, M. Deriche, and S. Nawaz, "X-OODM: Leveraging Explainable Object-Oriented Design Methodology for Multi-Domain Sentiment Analysis," *Comput. Mater. Continua*, vol. 82, no. 3, 2025, doi: 10.32604/cmc.2025.057359.

[19] H. Felzmann, E.F. Villaronga, C. Lutz, and A.T. Larrieux, "Towards transparency by design for artificial intelligence," *Sci. Eng. Ethics*, vol. 26, no. 6, pp. 3333–3361, 2020, doi: 10.1007/s11948-020-00276-4.

[20] A. Simkute, E. Luger, B. Jones, M. Evans, and R. Jones, "Explainability for experts: A design framework for making algorithms supporting expert decisions more explainable," *J. Responsible Technol.*, vol. 7, p. 100017, 2021, doi: 10.1016/j.jrt.2021.100017.

[21] F.D. Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint*, 2017, arXiv: 1702.08608, doi: 10.48550/arXiv.1702.08608.

[22] Z.C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[23] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019, doi: 10.1016/j.artint.2018.07.007.

[24] J. Nielsen, "Usability Engineering," in *Interactive Technologies*, Elsevier Science, 1994.

[25] T.J. McCabe, "A Complexity Measure," *IEEE Trans. Softw. Eng.*, vol. SE-2, no. 4, pp. 308–320, 1976, doi: 10.1109/TSE.1976.233837.

[26] K. Makhlouf, S. Zhioua, and C. Palamidessi, "On the applicability of machine learning fairness notions," *ACM SIGKDD Explor. Newslett.*, vol. 23, no. 1, pp. 14–23, 2021, doi: 10.1145/3468507.3468511.

[27] T. Gebru *et al.*, "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, pp. 86–92, 2021, doi: 10.1145/3458723.

[28] T. Huynh, N. Tsakalakis, A. Helal, S. Stalla, and L. Moreau, "Explainability-by-Design: A Methodology to Support Explanations," in *Decision-Making Systems*, pp. 1–12, 2022, doi: 10.48550/arXiv.2206.06251.

[29] J. Hsiao, H.H.T. Ngai, Q. Luyu, Y. Yang, and C. Cao, "Roadmap of Designing Cognitive Metrics for Explainable Artificial Intelligence (XAI)," *arXiv preprints*, 2021, arXiv: 2108.01737.

[30] T.J. McCabe and C.W. Butler, "Design complexity measurement and testing," *Commun. ACM*, vol. 32, no. 12, pp. 1415–1425, 1989, doi: 10.1145/76380.76382.

[31] M.K. S. Vadla, M.A. Suresh, and V.K. Viswanathan, "Enhancing product design through AI-driven sentiment analysis of Amazon reviews using BERT," *Algorithms*, vol. 17, no. 2, p. 59, 2024, doi: 10.3390/a17020059.