IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 September 2024, accepted 5 October 2024, date of publication 10 October 2024, date of current version 24 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3477553

RESEARCH ARTICLE

X-OODM: Explainable Object-Oriented Design Methodology

ABQA JAVED [1], MUHAMMAD SHOAIB [1], AND ABDUL JALEEL [2]
[1]Department of Computer Science, University of Engineering and Technology, Lahore 54890, Pakistan
[2]Department of Computer Science, Rachna College of Engineering Technology (RCET, GRW), University of Engineering and Technology, Lahore 52250, Pakistan

Corresponding author: Abqa Javed (abqa.javed@uet.edu.pk)

ABSTRACT In software applications and decision-making systems, the explainability features can be instrumental for explicating internal working, accountability, understanding, fairness, and interpretation of decisions, processes, and data. Conventional design methodologies like Object-Oriented Design Methodology (OODM) are proposed for web-based application development. OODM enables the reuse of code, quantification, and security at the design level. However, OODM did not provide the feature of introducing explainability in web-based decision-making systems, thus OODM is required to be modified. The present paper presents X-OODM with an added model to introduce the explainability feature. Design quality metrics for X-OODM are also proposed. The proposed methodology is validated through a case study involving different scenarios. In the first scenario, trustworthiness, fairness, transferability, and simulatability are implemented, resulting in an explainability level of 24 units. In addition to these components, in the second scenario, reliability, understanding, informativeness, and decomposability are involved with the previous parameters, having an explainability level of 34 units. In the third scenario, in addition to defined parameters, privacy awareness, accessibility, and algorithmic transparency components are also implemented, leading to the highest level of explainability 46 units compared to previous scenarios. A higher explainability level indicates that all aspects of web-based applications introduce explainability. This research can be extended to implement X-OODM for a real multi-domain sentiment analysis application.

INDEX TERMS Explainable, measurable, web-based application, object-oriented design, sentiment analysis.

I. INTRODUCTION

In the digital era, web-based systems are essential for many critical applications, including decision-making systems [1], which enable everything from e-commerce platforms to medical diagnoses [2]. These systems are client-server designs, built to manage complex interactions and an immense amount of data [3]. Traditional object-oriented design methodologies [4], such as OODM [5], Semantic Web object-oriented design methodology (SW-OODM) [6], reverse object-oriented design methodology (R-OODM) [7], and secure object-oriented design methodology (S-OODM). [8], provided solid frameworks to build these systems. However, these techniques must address the incorporation of explainability characteristics and capabilities into web-based decision-making systems [9].

Explainability is considered a valuable feature and a necessity for web-based decision-making systems [10]. It boosts the interpretations of decision-making systems and facilitates the stakeholders to understand the system-generated recommendations [11]. Therefore, explainable systems enhance the accountability of the system and incorporate compliance with regulatory standards [12], which is important in various domains including healthcare, finance, and education [13].

Despite the significance of the explainability, existing methodologies do not adequately address this need [14]. Traditional OODM methodologies have focused only on the functional and structural aspects of web-based systems, leaving a gap in the design phase which concerns how

The associate editor coordinating the review of this manuscript and approving it for publication was Asadullah Shaikh.

decisions are made and communicated to users [4]. This leads to a lack of transparency which creates trust issues, reduces user engagement, minimizes optimization opportunities, and potential operational inefficiencies [15].

Explainable Object-Oriented Design Methodology (X-OODM) is described in this paper as an extension of conventional OODM that incorporates explainability into the fundamentals of web-based system design. X-OODM aims to fill the gap by integrating models and components that enable transparency across the system's design. Implementing explainability in design improves the user experience and optimizes the decision-making process [16], which makes it more robust and understandable. In this methodology, explainability is incorporated in the design and analysis phases, making the systems provide clear, actionable, and understandable insights to the end-users. This approach is implemented in multidomain sentiment analysis to make complex and nuanced decisions more interpretable. Sentiment analysis has various applications for multiple domains [17]. X-OODM helps to accompany each sentiment classification by explaining the reasons for making the system more trustworthy and reliable.

In this research, a set of design quality metrics is introduced specifically for an explainable model that measures the effectiveness of X-OODM. These metrics are validated using the web-based application for multi-domain sentiment analysis which enhances the explainability of the system and quantifies it. The results demonstrate that incorporating the metrics early in design leads to transparency, reliability, functionally robustness, and user-friendlessness [18]. X-OODM addresses the need of explainability by presenting the significant advancement in the design level of web-based applications. Introducing the various X-OODM components including transparency, reliability, trustworthiness, and fairness into the web-based system, ensures the decision-making process is more justifiable, clear, and optimized for better outcomes. This methodology aligns technical excellence with explainability features for the development of web-based applications.

The article is arranged as follows. In the next section, we present the Literature Review. In Section III, the proposed work is presented and the quantification of the explainable model is described. Results and discussion are given in section IV. The paper's conclusion is presented in section V.

II. LITERATURE REVIEW

Various explainable approaches have been introduced to offer details about the inner workings, however they are all limited to the developer end. Existing approaches give explanations at the evaluation level, but no methodology is used to assess model performance at the design level. OODM is a design-level technique that uses object-oriented principles to construct web-based applications, ensuring complete flow before evaluation.

Liang and O'Grady [19] describe the design with objects (DwO) for the development of the design process model. The main aim of this approach is to provide the computability, reusability, and exchangeability of objects with similar and exchange data in modules. The dwO design process and its architecture were implemented to solve the problem of electronic assemblies' components. Shah [4] presented the object-oriented design methodology (OODM) used for web applications. Many design-level techniques have been proposed for the web application, but they did not implement the complete software development lifecycle. OODM is constructed on the waterfall software development life cycle with the integration of web operations in objects for applications. Ghani et al. [7] described the reverse object-oriented design methodology (R-OODM) which extracts the web application design using design phase models of OODM. XML schema is mapped with an object-oriented database and translated the schema into a graph.

Kiewkanya et.al [20] also worked on the maintainability model in the object-oriented design model using various techniques. Modifiability and understandability parameters are the major concerns in imposing the maintainability performed on experimental data. Metrics discriminant technique implemented to identify the relations levels of maintainability and structural complexity. Moreover, weighted sum and predicted level techniques are outputs of the above levels and convert them into scores.

Arshad [8] extend the OODM methodology by incorporating the security paradigms at the design level. The existing methodology provides the design and analysis phases but does not tackle the security issues in the OODM for web applications. Therefore, in the secure object-oriented design methodology (S-OODM) security model at the design level is introduced which is then evaluated through design-level metrics. Kadam and Joshi [21] discussed the quantification of security to minimize the vulnerabilities to handle security levels. Security metrics helped the designers enhance the security at the design level by measuring it. They follow the guidelines to check the security level and enhance it accordingly.

Jangra and Dua [22] used the object-oriented database for the health case management system. The designed system provides the requirements including managing data views and schema views, suitable ways to handle the temporal information, and enhancement of multimedia data. Mohapatra et.al [23] secured the sensitive information using the object-oriented methodology in the cloud healthcare system. A single window interface is provided to facilitate the citizens. They improved the modularity and flexibility of the cloud healthcare system design through object-oriented principles which enhanced the collaboration and communication between various components due to emphasis on behavioral aspects. Nwokoro [24] also described the object-oriented analysis and design methodology for the university management system. Due to the relationships of objects, data

TABLE 1. Comparison with existing studies of OODM.

Ref.	Object-Oriented Design	Web-Based Applications	Multi-Domain Sentiment Analysis	Explainability	Quantification
[19]	✓	✓	×	×	×
[4]	✓	✓	×	×	×
[7]	✓	✓	×	×	×
[20]	✓	✓	×	×	✓
[8]	✓	✓	×	×	✓
[21]	✓	✓	×	×	✓
[22]	✓	✓	×	×	×
[23]	✓	✓	×	×	×
[24]	✓	✓	×	×	×
[25]	✓	×	×	✓	×
[26]	✓	✓	×	×	✓
Proposed Work	✓	✓	✓	✓	✓

retrieval is more efficient which enhances the functionality of the system.

Arulprakash and Martin [25] describe the object-oriented methodology for neural representation and its applications for explainable AI. The feature importance techniques are implemented through objects which help to identify the correlation between weight and loss distribution. Extensibility property introduced for business parameters using an object-oriented design that calculates the new loss functions. Geyer et.al [26] present the unified approach of component-based machine learning methods using explainable AI. As these models have black boxes in nature, to get insights into these models' component-based approach is implemented with systems engineering. The explanation of these models is evaluated through qualitative and quantitative methods which compare the results of the component-based model with the white-box simulation results.

Table 1 compares the proposed work to the current literature on OODM. No study uses the object-oriented design technique for web-based multi-domain sentiment analysis. The quantification of explainable models at the design level is also lacking in the literature, considering the established parameters. To make a web-based multi-domain sentiment analysis program transparent and fair to users, the quantification of explainable models must be integrated into the object-oriented design.

III. PROPOSED WORK

A. X-OODM

X-OODM is an extension of OODM [4] that is specifically designed for web-based applications, with a focus on explainability. X-OODM introduces explainability in both the analysis and design phases, which leads to clarity and transparency in decision-making. X-OODM distinguishes from conventional methods during the analysis phase by describing the identification and description of system needs, making the logic behind these requirements more understandable. The aspects of explainability also have an impact on several existing design models, such as informational, navigational, operational, component, and user interface models. Figure 1 depicts the complete framework of an explainable model which emphasizes the impact of explainability on present OODM. This integration not only improves the system's interpretability but allows users to confidently grasp the complexities of web-based applications.

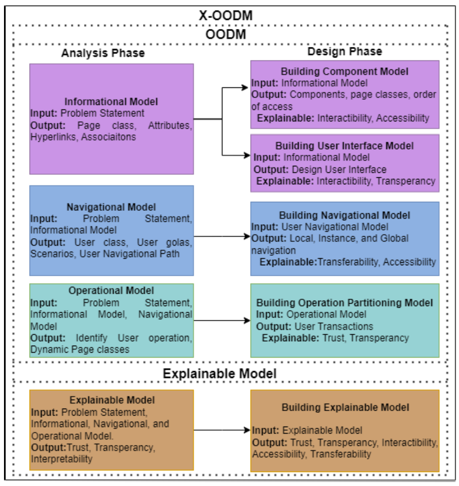

FIGURE 1. Integration of explainable model in OODM.

Figure 1 illustrates the interconnectivity of the analysis as well as the design phase in our methodology. In this schematic representation, each model created during the analysis phase is integrated as an input into the subsequent design process. Our proposed methodology X-OODM involves enhancing existing design models by including explainability and then upgrading them through the design of transparent web-based applications. The designed model incorporates the explainability in the web-based applications by providing the explainable output to the user.

In Figure 2, the Explainable Model is illustrated, introducing four models aimed at integrating explainability into web-based applications. Each model has various parameters and components to integrate the explainability at the design level. Numerous studies [27], [28], and [29] highlight that explainability plays a crucial role in organizational compliance, offering insight into internal operations in a privacy-conscious and consistent manner.

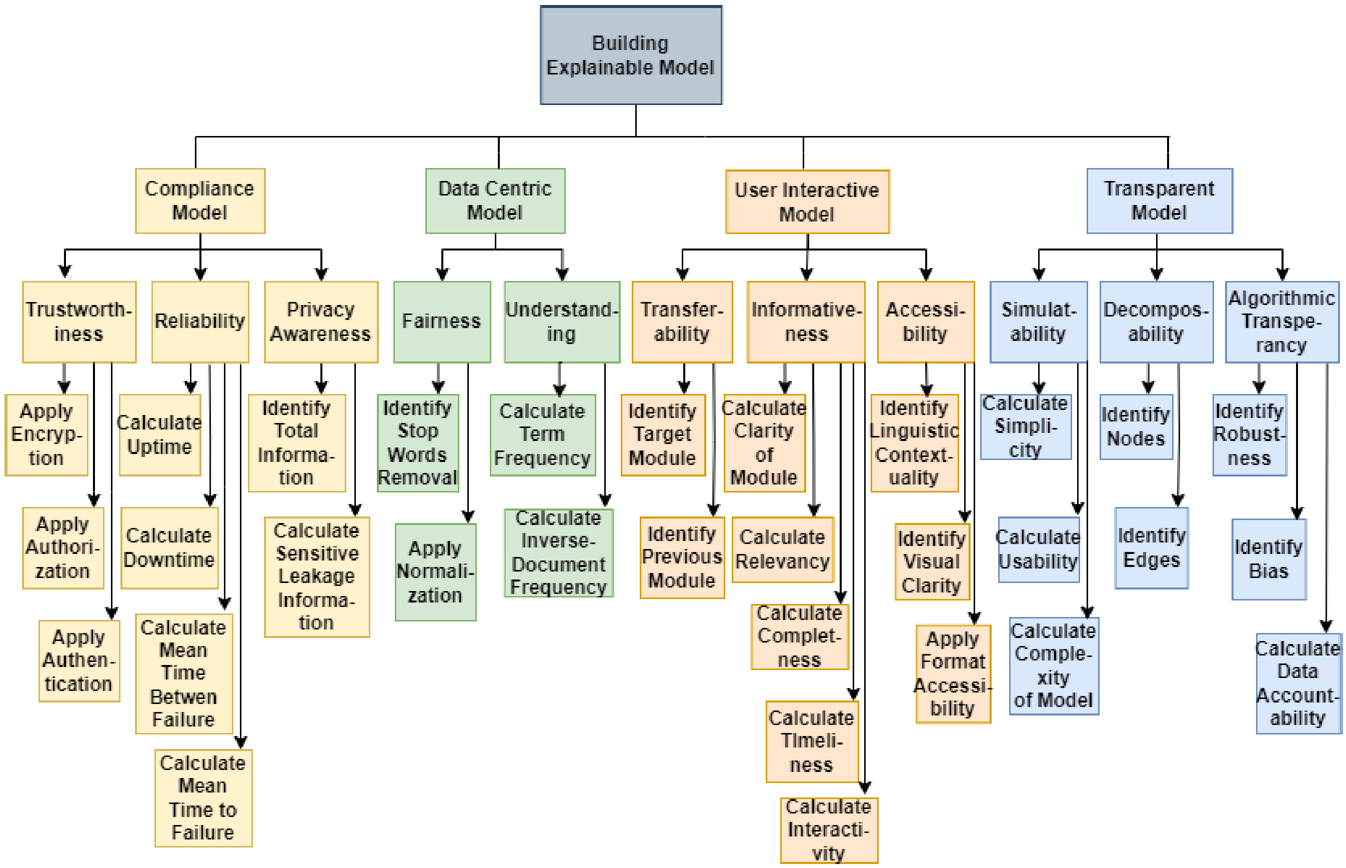

FIGURE 2. Our proposed structure of building explainable model for X-OODM.

B. PROPOSED DESIGN QUALITY METRICS

Quantifying the explainable model presented in Figure 3 involves evaluating its qualities, such as clarity, transparency, comprehensibility, and interpretability. This method provides a framework for assessing how well the model conveys complex information, that helps to understand them [30]. By providing numerical values or metrics to these dimensions, individuals may objectively analyze the model's performance, compare different models, and certify the model's compliance with user expectations and regulatory requirements. In a nutshell, quantification provides a mechanism to assess the explainable model's quality and efficacy, allowing for informed decision-making and continual improvement to maintain its usability and dependability in delivering essential data explicitly. X-OODM defines the various components to introduce explainability in the framework based on the user requirements for web-based applications. These components are converted into the interaction graph of 'G' given in Figure 4. The basic components such as trustworthiness, reliability, and privacy awareness provide a trustworthy application to the user which comes under the compliance model. Data security and the details of data are provided to the users by introducing the fairness and understanding components in the data-centric model. The user interactive model directly links with the front of the application which has transferability, informativeness, and accessibility components. Transparent models provide the components such as simulatability, decomposability, and algorithmic transparency. These models merge the input and send it to the explainable model which provides the overall explainability in the web-based applications.

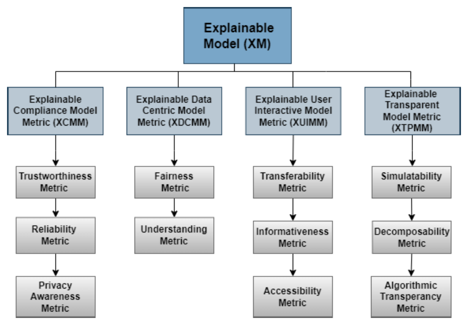

FIGURE 3. Quantification of explainable model.

C. SCHEMA REPRESENTATION OF THE EXPLAINABLE MODEL

For the explainable schema interpretation based on the defined abstract schema graph in Figure 4, we defined the following terms in each layer of the model:

E_l: Low-End Level
F_m: Component Level
G_n: Basic Model Level

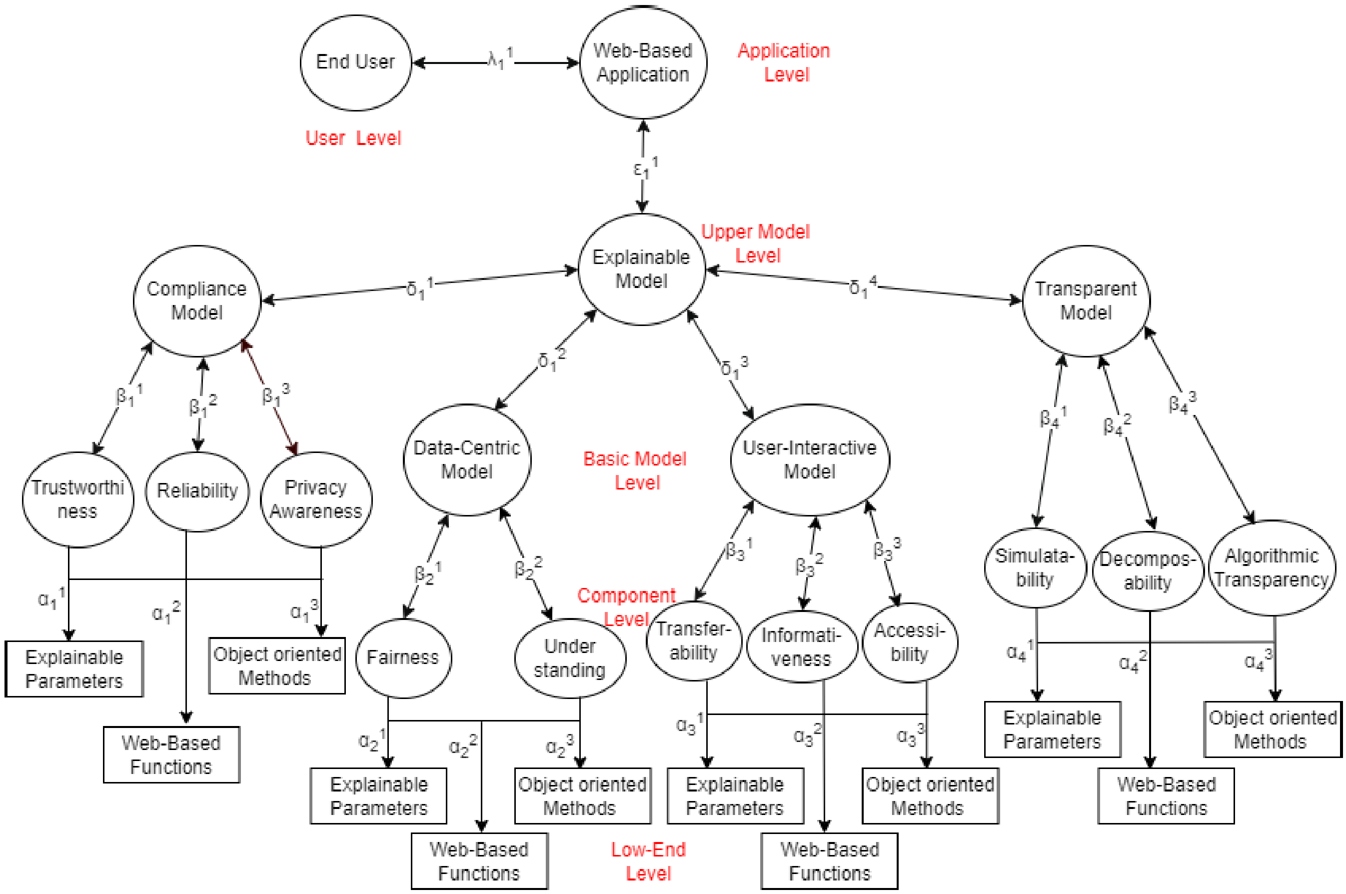

FIGURE 4. Abstract graph for the components of explainable model.

H_o: Upper Model Level
I_p: Application Level
J_q: User Level

The integer numbers such as l, m, n, o, p, and q denote the components and models that are used to provide explainability at the design level in web-based applications. Interaction Graph 'G' is used to define the measurements related to the design quality of explainable web-based applications. This graph further presented its vertices V(G), given in

$$V(G) = \{E_u\} \cup \{F_v\} \cup \{G_w\} \cup \{H_x\} \cup \left\{I_y\right\} \cup \{J_z\} \qquad (1)$$

where $1 \le u \le l$, $1 \le v \le m$, $1 \le w \le n$, $1 \le x \le o$, and $1 \le y \le p$

the terms u, v, w, x, and y are integer values used to denote the index numbers of the Low-End level, Components level, Basic Model level, Upper Model level, Application Level, and User Level, respectively.

The bidirectional edges in the abstract graph denote the components that have more than one edge originating from the vertex. Therefore, the edges are assigned a unique symbol that represents the different vertex.

$$ComponentLevel(F\_m) \Leftrightarrow Low - EndLevel \_\_\_\_\_\_ \alpha_v^{(i\_v)}$$
$$BasicModelLevel(G\_n) \Leftrightarrow ComponentLevel \_\_\_\_\_\_ \beta_w^{(i\_w)}$$
$$UpperModelLevel(H\_o) \Leftrightarrow BasicModelLevel \_\_\_\_\_\_ \gamma_x^{(i\_x)}$$
$$ApplicationLevel(I_p) \Leftrightarrow UpperModelLevel \_\_\_\_\_\_ \epsilon_y^{(i\_y)}$$
$$UserLevel(J\_q) \Leftrightarrow ApplicationLevel \_\_\_\_\_\_ \lambda_z^{(i\_z)}$$

The integers v, w, x, y, and z are the subscripts of the edges which denote the vertex number of the respective edge such as α, β, γ, ϵ, and λ.

Where $1 \le v \le m$, $1 \le w \le n$, $1 \le x \le o$, $1 \le y \le p$, and $1 \le z \le q$.

Whereas the edge number of each vertex F_m, G_n, H_o, I_p, and J_q used the subscript integers

such as i_v, i_w, i_x, i_y, and i_z, which acts as a counter of the multiple edges.

The defined edges are revolved in the following order:

$\alpha_v^{(i_v)}$: $\alpha_1^1, \alpha_1^2, \alpha_1^3, \ldots\ldots\ldots\ldots \alpha_1^{(i_1)}, \alpha_2^1, \alpha_2^2, \alpha_2^3, \ldots\ldots\ldots\ldots \alpha_2^{(i_2)}, \alpha_3^1, \alpha_3^2, \alpha_3^3, \ldots\ldots\ldots\ldots \alpha_3^{(i_3)}$

::::

$\alpha_v^1, \alpha_v^1, \alpha_v^1, \cdots\cdots\cdots \alpha_v^{(i_v)}$

These edges are measured as the weights when the data is sent from each component to the respective model during single interaction activation.

OODM creates various classes and objects to design web-based applications. This work incorporates the explainability classes and objects into the X-OODM. Explainable components are introduced into the existing OODM to make the design of the web-based applications explainable for the end users. These components provide the output that is assigned to the related model and make the web-based applications transparent and fair for the users. They get insights into the basic operations and details of the data which enhances the trust of the users on the system. In terms of interaction edges between the distinct vertices of the abstract graph, explainable components are described as follows:

$$E_x = \left\{\left(\alpha_i^{i'} \varepsilon \in E_x\right) \subseteq \alpha_v^{(i_v)}\right\} \cup \left\{\left(\beta_j^{j'} \varepsilon \in E_x\right) \subseteq \beta_w^{(i_w)}\right\} \cup \left\{\left(\delta_k^{k'} \varepsilon \in E_x\right) \subseteq \delta_k^{(i_k)}\right\} \cup \left\{\left(\varepsilon_l^{l'} \varepsilon \in E_x\right) \subseteq \varepsilon_l^{(i_l)}\right\} \cup \left\{\left(\lambda_m^{m'} \varepsilon \in E_x\right) \subseteq \lambda_m^{(i_m)}\right\} \quad (2)$$

Various explainable models are defined to enhance the trust of users on web-based applications which are further categorized into the subcomponents that are represented in Figure 4.

D. EXPLAINABLE COMPLIANCE MODEL METRIC (XCMM)

In the realm of web-based applications, incorporating a compliance model is critical to ensure consistency of legal, ethical, and industry-specific rules. By establishing a compliance model, developers provide a framework that regulates the application's behavior and operations, protecting user data while adhering to privacy and security guidelines. This strategy not only reduces the legal obligations associated with non-compliance but also increases user trust and confidence in the application's integrity. Furthermore, compliance with rules is a necessity for multidomain sentiment analysis systems, where data from all related fields are given to the model for processing. Moreover, integrating a compliance model might provide a competitive advantage by indicating the application's consistency with ethical standards and data protection, attracting users who value privacy and security.

1) TRUSTWORTHINESS

It includes characteristics such as accountability and ethical behavior to ensure that users can trust the system's outputs and operations. Users require strong security measures to protect their information from unauthorized access, breaches, and malicious activity. Trustworthiness can be achieved by incorporating various factors that include secured authentication processes, sensitive data encryption, protection against common attacks, and regular security upgrades.

The trustworthiness metric is defined in (3) which defines the parameters that directly impact the security of the system.

Encryption Metrics $= \in_i$

Authentication Metrics $= \forall_i$

Authorization Metrics $= \partial_i$

$$Trustworthiness\ Metrics = \sum_{i=1}^{j} (\in_i + \partial_i + \forall_i) \in \alpha_i^{i'} \quad (3)$$

The impact of trustworthiness is calculated in (4).

$$Average\ Trustworthiness\ Metrics = \frac{1}{n} \sum_{i=1}^{j} (\forall_i + \partial_i + \in_i) \in \alpha_i^{(i')} \quad (4)$$

2) RELIABILITY

Reliability is a critical aspect of system performance, particularly in the context of web-based applications. It focuses specifically on the consistency and accuracy of the system's outputs. A reliable system consistently produces accurate results under varying conditions, without significant fluctuations or errors. It can be achieved by calculating the various metrics including uptime, availability, and mean time between failures (MTBF), Mean time to failure (MTTF), and effective error handling to ensure a smooth and uninterrupted user experience.

The reliability metric of the system is defined in (5) which presents the factors for a reliable web-based system at the user end.

Uptime Metrics $= \cup_i$

Downtime Metrics $= D_i$

Failure Metrics $= F_i$

Availability $= \frac{\sum_{i=1}^{j} \cup_i}{\left(\sum_{i=1}^{j} \cup_i + \sum_{i=1}^{j} D_i\right)} \times 100$

Mean time between failure (MTBF) $= \frac{\sum_{i=1}^{j} \cup_i}{\sum_{i=1}^{j} F_i}$

Mean time to failure (MTTF) $= \frac{\sum_{i=1}^{j} \cup_i}{\sum_{i=1}^{j} F_i + 1}$

$$ReliabilityMetrics = \left[\sum_{i=1}^{j} \left\{\frac{\cup_i}{\cup_i + D_i}\right\} + \frac{\cup_i}{F_i} + \frac{\cup_i}{F_i + 1}\right] \in \alpha_i^{i'} \quad (5)$$

The impact of reliability over explainability can be calculated using (6).

$$AverageReliabilityMetrics = \frac{\left[\sum_{i=1}^{j} \left\{\frac{\cup_i}{\cup_i + D_i}\right\} + \frac{\cup_i}{F_i} + \frac{\cup_i}{F_i + 1}\right]}{n} \in \alpha_i^{i'} \quad (6)$$

3) PRIVACY AWARENESS

Integrating privacy into web-based apps allows them to handle sensitive data while preserving user confidence. End users are required to keep their personal information secret. They demand access to privacy settings, a clear explanation of data collecting for the model, and considering data protection standards. Calculating privacy in explainable web-based multi-domain sentiment analysis involves assessing the amount to which user data is secured and maintained secret during the analysis. The privacy of the system may be accomplished by measuring the information leakage from the entire information provided in (7).

Sensitive Leaked Information Metrics $= S_i$

Information Metrics $= I_i$

$$\textit{Privacy Awareness Metrics} = \left[\sum_{i=1}^{j}\left\{1-\left(\frac{S_i}{I_i}\right)\times 100\right\}\right]\in \alpha_i^{i'} \tag{7}$$

In the compliance model, various components are implemented to ensure the security and privacy of the web-based applications for multi-domain sentiment analysis. The above-mentioned metrics are used for the smooth integration of the compliance model in an explainable model.

$$\begin{aligned} XCM = &\left[\sum_{i=1}^{j}\{(\forall_i+\partial_i+\in_i)\}\right]\in\alpha_i^{i'} \\ &+\left[\sum_{i=1}^{j}\left\{\frac{\cup_i}{\cup_i+D_i}+\frac{\cup_i}{F_i}+\frac{\cup_i}{F_i+1}\right\}\right]\in\alpha_i^{i'} \\ &+\left[\sum_{i=1}^{j}\left\{1-\left(\frac{S_i}{I_i}\right)\times 100\right\}\right]\in\alpha_i^{i'} \end{aligned} \tag{8}$$

The XCM metrics (8) are used to measure compliance in web-based applications. These metrics are summed up in (9) to measure the overall impact of the compliance model in the explainable model.

$$\textit{Overall Impact of XCM Metrics} = \sum_{i=1}^{j} XCM_i \tag{9}$$

The average impact of the XCM metrics is calculated in (10).

$$\textit{Average Impact of XCM Metrics} = \frac{\sum_{i=1}^{j} XCM_i}{n} \tag{10}$$

By incorporating all the components of the compliance model, we implement all the rules and regulations in the web-based application through an explainable model.

E. EXPLAINABLE DATA-CENTRIC MODEL METRIC (XDCMM)

In the context of explainable models for web-based applications in multidomain sentiment analysis, the data-centric model focuses on data processing to allow for transparent and interpretable sentiment analysis across domains. This paradigm emphasizes the clarity and comprehensibility of sentiment analysis data, ensuring that users understand the analysis' inputs and outcomes. It includes data collection, preprocessing, feature extraction, and representation procedures that improve the interpretability of sentiment analysis results.

1) FAIRNESS

Fairness of data assures that the data utilized for analysis or modeling lacks the strength of biases or inequalities that might disproportionately affect groups or categories. Web-based systems require users to provide a fair and unbiased representation of datasets. Normalization methods are used in textual data preparation to ensure fairness by normalizing word representations and decreasing possible biases caused by variances in case sensitivity. It includes balanced data across different domains to avoid under or over-representing data. The defined metrics of fairness are the measures to ensure the data that is passed for the multi-domain sentiment analysis is well-balanced and not case-sensitive. The fairness metrics can be defined in (11).

Searching Metrics $= \delta_i$

Cleaning Metrics $= \mathbf{l}\delta_i$

Insignificant Words Removal Metrics $= \left\{\sum_{i=1}^{j}(\delta_i \times \mathbf{l}\delta_i)\right\} \in \alpha_i^{i'}$

Normalization Metrics $= \left\{\sum_{i=1}^{j}(\text{alphabet\_change})\right\} \in \alpha_i^{i'}$

$$\textit{Fairness Metrics} = \sum_{i=1}^{j}\{(\delta_i\times \mathrm{l}\delta_i)+\text{alphabet\_change}\}\in\alpha_i^{i'} \tag{11}$$

The average Impact of fairness can be calculated in (12).

$$\textit{Average ratio of the Fairness Metrics} = \frac{\left(\sum_{i=1}^{j}\{(\delta_i\times \mathrm{l}\delta_i)+\text{alphabet\_change}\}\in\alpha_i^{i'}\right)}{n} \tag{12}$$

2) UNDERSTANDING

The system's behavior and decisions are straightforward and comprehensible. An understandable system allows users to analyze and interpret its outputs, increasing trust and encouraging collaboration between humans and technology. Users require insights into the significance of various attributes or phrases in determining sentiment analysis results. Feature significance analysis tools, such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings, assist users in identifying which words or phrases represent the most related information.

TF-IDF is calculated in (13) which provides insights into the important features that directly impact the overall performance of the sentiment analysis applications.

Term Metrics $= \tau_i$

Document Metrics $= d_i$

Each term in the document $= \delta(\tau, \tau_i)$

Number of documents having term $= \delta(\tau, d_i)$

Term-Frequency Metrics $= \frac{\sum_{i=1}^{j}\delta(\tau,\tau_i)}{\tau_i}$

Inverse-Document-Frequency Metrics $= \log\frac{\left(\sum_{i=0}^{j} d_i\right)}{\delta(\tau,d_i)}$

$$\textit{Understanding Metrics}(TF-IDF) = \left[\sum_{i=1}^{j}\left\{\left(\frac{\sum_{i=1}^{j}\delta(\tau,\tau_i)}{\tau_i}\right)\times\log\frac{\left(\sum_{i=0}^{j} d_i\right)}{\delta(\tau,d_i)}\right\}\right]\in\alpha_i^{i'} \tag{13}$$

Calculate the average of the understanding metrics by using (14).

$$\textit{Average ratio of Understanding Metrics} = \frac{\left[\sum_{i=1}^{j}\left\{\left(\frac{\sum_{i=1}^{j}\delta(\tau,\tau_i)}{\tau_i}\right)\times\log\frac{\left(\sum_{i=0}^{j}d_i\right)}{\delta(\tau,d_i)}\right\}\right]}{n} \in \alpha_i^{i'} \tag{14}$$

In web-based applications for multidomain sentiment analysis, data is the most important component. The explainability model ensures to provision insights of into the data to the end user by introducing the data-centric model. It provides the fairness and understanding of the data by extracting the most relevant features from the dataset that directly impact the output. The design metrics (11) and (13) are used to get the details of the data and implement it in the explainable model defined in (15).

$$XDCM = \left[\sum_{i=1}^{j}\{(\delta_i \times \mathrm{l}\delta_i) + \text{alphabet\_change}\}\right] + \left[\sum_{i=1}^{j}\left\{\left(\frac{\sum_{i=1}^{j}\delta(\tau,\tau_i)}{\tau_i}\right)\times\log\frac{\left(\sum_{i=0}^{j}d_i\right)}{\delta(\tau,d_i)}\right\}\right] \in \alpha_i^{i'} \tag{15}$$

These metrics are designed to measure the impact of the data-centric model in the explainable model in (16).

$$\textit{Overall Impact of XDCM Metrics} = \sum_{i=1}^{j}\mathrm{XDCM}_i \tag{16}$$

Below defined (17) describes the average impact of the model.

$$\textit{Average Impact of XCM Metrics} = \frac{\sum_{i=1}^{j}\mathrm{XDCM}_i}{n} \tag{17}$$

By implementing the data-centric model, it is assumed that the explainable model gives data insights to the user end and presents the usage of data as an output to avoid bias.

F. EXPLAINABLE USER INTERACTIVE MODEL METRIC (XUIMM)

In the context of multidomain sentiment analysis in web-based applications, the User Interactive Model develops as a fundamental method that includes transferability, informativeness, and accessibility. This technique integrates individuals into the sentiment analysis process by implementing domain-specific information and context through interactive interfaces and feedback mechanisms.

Furthermore, the User Interactive Model intends to improve the usefulness of sentiment analysis results by providing users with extensive insights into sentiment patterns and factors through categorization. Using visualization tools and explanation interfaces, users may get explicit explanations for sentiment predictions, which helps them grasp the analytic results. Furthermore, the model stresses accessibility by offering user-friendly interfaces, simple visualization tools, and configurable parameters, guaranteeing that sentiment analysis findings are available to users of any level of knowledge and preferences.

1) TRANSFERABILITY

Transferability increases the system's ability to communicate information and insights from automated analysis to users. It provides attributes that allow users to provide domain-specific information or context to improve the sentiment analysis model and enable it to effectively generalize across domains. The transferability metrics are achieved by calculating the performance of the target module and the previous module for the same task which helps to determine whether it accurately transfers the data between modules or not. (18) presents the transferability metrics and the average impact of the transferability is achieved in (19).

Target Metrics $= \phi_i$
Previous Metrics $= \rho_i$

$$\textit{Transferability Metrics} = \left[\sum_{i=1}^{j}\left\{\frac{(\phi_i - \rho_i)}{\phi_i}\times 100\right\}\right] \in \alpha_i^{i'} \tag{18}$$

$$\textit{Average ratio of Transferability Metrics} = \frac{\left[\sum_{i=1}^{j}\left\{\frac{(\phi_i-\rho_i)}{\phi_i}\times 100\right\}\right]}{n} \in \alpha_i^{(i')} \tag{19}$$

The high transferability to the next module score implies that the current module successfully applies the information, or features attained in the previous module to increase performance on the target task.

2) INFORMATIVENESS

In the user interaction model, informativeness guarantees that the interface successfully delivers information to users in clear, relevant, and efficient ways. Its goal is to increase users' understanding, trust, and confidence in the system's operations and ideas. The interface should provide information relevant to the user's specific context, tasks, preferences, and objectives. It also delivers timely updates and notifications through the interface, particularly for time-sensitive information or changes in system behavior. Furthermore, informativeness aims to provide actionable insights that users may use to improve decision-making processes and produce meaningful interactions with the sentiment analysis system, therefore enhancing overall satisfaction with the application.

The various components are involved to provide information in the user interactive model. The informativeness metrics are defined in the (20).

Clarity Metrics $= C_i$
Relevance Metrics $= \pi_i$
Completeness Metrics $= \theta_i$
Timeliness Metrics $= T_i$

Interactivity Metrics $= N_i$

$$Informativeness\ Metrics = \sum_{i=1}^{j}(C_i+\pi_i+\theta_i+T_i+N_i) \in \alpha_i^{i\prime} \tag{20}$$

The average impact of the informativeness in the user interactive model is calculated in (21).

$$Average\ ratio\ of\ Informativeness\ Metrics = \frac{\left[\sum_{i=1}^{j}(C_i+\pi_i+\theta_i+T_i+N_i)\right]}{n} \in \alpha_i^{i\prime} \tag{21}$$

3) ACCESSIBILITY

The goal of accessible sentiment analysis data presentation is to give users clear and simple explanations, regardless of their data analysis expertise. This is accomplished by using simple language to describe sentiment analysis results and avoiding advanced terminology that may be challenging to understand. In addition, the system includes a variety of customization possibilities, allowing users to modify the presentation of data to their tastes and needs. The sentiment analysis system ensures that users can interact with and derive meaningful insights from data by prioritizing accessibility in language, visualizations, and formats, regardless of their level of data analysis skills. This method increases integration and allows users to make informed decisions based on sentiment analysis results which helps to enhance the system's usefulness.

Linguistic Contextuality Metrics $= L_{\emptyset_i}$
Visual Clarity Metrics $= Vc_i$
Format Accessibility Metrics $= F\nabla_i$

$$Accessibility\ Metrics = \sum_{i=1}^{j}(L_{\emptyset_i}+V_{c_i}+(F\nabla)_i) \in \alpha_i^{(i')} \tag{22}$$

(22) describes the complete accessibility metrics having the parameters of relevancy of the language, visual clarity, and usability of the format. The average ratio of the accessibility metrics in (23) is defined.

$$Average\ Ratio\ Of\ Accessibility\ Metrics = \frac{\left[\sum_{i=1}^{j}(L_{\emptyset_i}+V_{c_i}+(F\nabla)_i) \in \alpha_i^{i\prime}\right]}{n} \tag{23}$$

The explainability is introduced through a user interactive model which allows the users to directly interact with the interface and then get the details of the processing. It implies a significant impact on the web-based applications to make it explainable for multidomain sentiment analysis. The effect of the user interactive model is measured through various components using metrics. The metrics of (18), (20), and (22) are added up in (24) to check the overall impact on the model.

$$\begin{aligned} XUIM = &\left[\sum_{i=1}^{j}\left(\frac{(\phi_i-\rho_i)}{\phi_i}\times 100\right)\right] \in \alpha_i^{i} \\ &+\left[\sum_{i=1}^{j}(C_i+\pi_i+\theta_i+T_i+N_i)\right] \in \alpha_i^{i\prime} \\ &+\left[\sum_{i=1}^{j}(L_{\emptyset_i}+V_{c_i}+(F\nabla)_i)\right] \in \alpha_i^{(i')} \end{aligned} \tag{24}$$

The overall impact of the XUIM is measured in (25) and (26) shows the average impact of the model.

$$Overall\ Impact\ of\ the\ XUIM\ Metrics = \sum_{i=1}^{j}\mathrm{XUIM}_i \tag{25}$$

$$Average\ Impact\ of\ the\ XUIM\ Metrics = \frac{\sum_{i=1}^{j}\mathrm{XUIM}_i}{n} \tag{26}$$

Transferability, informativeness, and accessibility components are defined to make the user-interactive model explainable for users.

G. EXPLAINABLE TRANSPARENT MODEL METRIC (XTPM)

Transparent models are essential to develop to gain the trust of users by providing them insight into sentiment analysis. Transparent models, as compared to complex models that work as black boxes, provide insight into their ideas. This enables users to understand the reasons behind certain sentiment predictions. To achieve transparency, the model should be developed using various simple and understandable methodologies including decision trees and logistic regression. These methodologies enable users to observe and explain the elements that impact sentiment analysis across several domains.

Furthermore, transparent models can explain predictions about sentiments. They give users insights into the important parameters or components that influence sentiment categorization. This transparency enhances trust and confidence by allowing users to apply their knowledge to evaluate and verify the results.

1) SIMULATABILITY

Simulatability in a transparent model is required for users to grasp and replicate the model's decision-making process. It is required to understand the logic that underlies the model's predictions by following its stages in a way such as human cognition. Simulatability improves user confidence in the model's predictions by allowing them to check the accuracy and fairness of the outcomes. It also enables users to uncover model biases and make educated decisions based on analysis results. Additionally, simulatability is critical for ensuring a model's transparency and interpretability, allowing users to make better use of the analytic results.

Simplicity Score Metrics $= SS_i$
Usability Score Metrics $= US_i$
Model's Complexity Metrics $= MC_i$

$$Simulatability\ Metrics = \sum_{i=1}^{j}(SS_i+US_i+MC_i) \in \alpha_i^{(i')} \tag{27}$$

Simulatability in the transparent model is calculated by introducing the parameters including simplicity of the model, usability of the model, and working complexity of the model presented in (27). The average ratio of simulatability is calculated in (28).

$$\textit{Average ratio of Simulatability Metrics} = \frac{\sum_{i=1}^{j}(SS_i + US_i + MC_i)}{n} \in \alpha_i^{(i')} \tag{28}$$

2) DECOMPOSABILITY

Decomposability enables users to understand a model's complex workings by breaking it down into smaller components. Users can explore the contribution of each component to the model's predictions by splitting them into individual characteristics, specifications, or decision rules. The users must be integrated into a modular system with well-defined components that can be easily handled. The model should be able to divide the system into smaller, more manageable pieces. Cyclometric complexity is used to assess the explainable model's decomposability. It is often used to assess code quality, indicate suitable refactoring regions, and estimate testing efforts.

Edges Metrics $= \in_i$
Nodes Metrics $= \cap_i$
Connected Components $= CC_i$
Cyclometric Metrics $= \in_i - \cap_i + 2\mathrm{CC}_i$

$$\textit{Decomposability Metrics} = \sum_{i=1}^{j}(\in_i - \cap_i + 2\mathrm{CC}_i) \in \alpha_i^{(i')} \tag{29}$$

Decomposability metrics are calculated by implementing the cyclometric metrics in (29) and (30) describe the average ratio of these metrics.

$$\textit{Average ratio of Decomposability Metrics} = \frac{\sum_{i=1}^{j}(\in_i - \cap_i + 2\mathrm{CC}_i)}{n} \in \alpha_i^{(i')} \tag{30}$$

3) ALGORITHMIC TRANSPARENCY

Algorithmic transparency in an explainable model is essential to developing trust in its predictions. This concept indicates making the algorithm's inner workings and decision-making processes clear and accessible to users. Algorithmic transparency can be implemented in several methods, including explanations for prediction, incorporating interpretable algorithms that exhibit model activity with interactive tools, and extensively documenting model operations. Furthermore, by emphasizing algorithmic transparency, explainable models can increase their usability and efficiency across an extensive number of applications, allowing users to make more informed decisions based on a better knowledge of the model's behavior. Algorithmic transparency is calculated in (31).

Robustness Metrics $= R\mu_i$
Bias Detection Metrics $= BD_i$
Data Accountability Metrics $= D\forall_i$

$$\textit{Algorithmic Transparency} = \sum_{i=1}^{j}(R\mu_i + BD_i + D\forall_i) \in \alpha_i^{(i')} \tag{31}$$

The average ratio of algorithmic transparency has an impact on the overall transparent model defined in (32).

$$\textit{Average ratio of Algorithmic Transparency} = \frac{\sum_{i=1}^{j}(R\mu_i + BD_i + D\forall_i)}{n} \in \alpha_i^{(i')} \tag{32}$$

In the transparent model of explainability, different components are discussed that have various parameters to introduce explainability in web-based applications. The above-mentioned metrics of XTPM are used to measure transparency in explainability. The overall design metrics are combined in (33).

$$\begin{aligned} XTPM = &\sum_{i=1}^{j}(SS_i + US_i + MC_i) \in \alpha_i^{(i')} \\ &+ \sum_{i=1}^{j}(\in_i - \cap_{i+2}\mathrm{CC}_i) \in \alpha_i^{(i')} \\ &+ \sum_{i=1}^{j}(R\mu_i + BD_i + D\forall_i) \in \alpha_i^{(i')} \end{aligned} \tag{33}$$

The overall impact of the XTPM in the explainable model is measured in (34) average impact is calculated in (35).

$$\textit{Overall Impact of XTPM} = \sum_{i=1}^{j}\mathrm{XTPM}_i \tag{34}$$

$$\textit{Average Impact of the XTPM Metrics} = \frac{\sum_{i=1}^{j}\mathrm{XTPM}_i}{n} \tag{35}$$

Transparency emphasizes accountability and allows users to understand and evaluate model behavior, which leads to enhanced decision-making knowledge and reduced biases. Transparent models also help with the ethical usage of AI systems by encouraging transparency and accountability in algorithmic decision-making.

We define the metrics of individual components of the explainable model and merge them into the respective model. These design metrics create the interaction between the components including trustworthiness, reliability, privacy, transferability, and accessibility of the explainable model which helps to provide the explainability at the design level of the web-based applications. Given defined models such as the Compliance model, Data-Centric model, User Interactive model, and Transparent model have a direct impact on the explainability of the web-based application as shown in (36) and the average is calculated in (37).

Compliance Metrics $= \sum_{i=1}^{j}\mathrm{XCM}_i$
Data Centric Metrics $= \sum_{i=1}^{j}\mathrm{XDCM}_i$

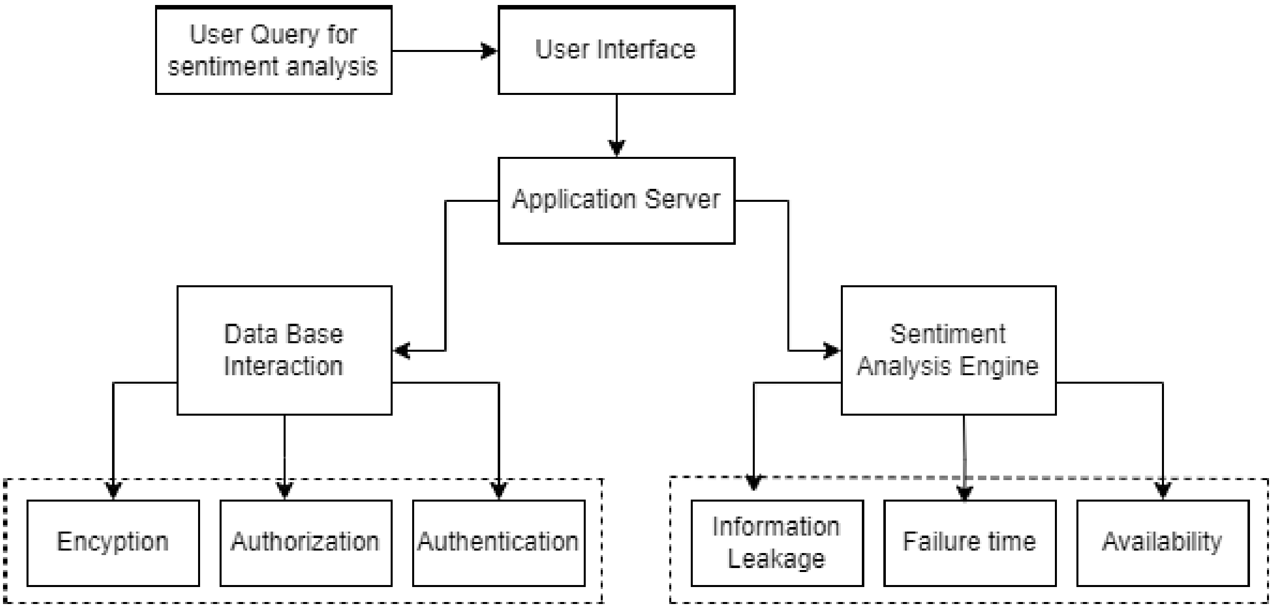

FIGURE 5. Web-based application for multidomain sentiment analysis.

User Interactive Metrics $= \sum_{i=1}^{j} \mathrm{XUIM}_i$
Transparent Metrics $= \sum_{i=1}^{j} \mathrm{XTPM}_i$

$$Overall\ Impact\ of\ Explainable\ Model\ Metrics(XDCUITM) = \sum_{i=1}^{j} (XCM_i + XDCM_i + XUIM_i + XTPM_i) \quad (36)$$

Average Impact of Explainable Model Metrics (XDCUITM) =

$$\frac{\sum_{i=0}^{j} (XCM_i + XDCM_i + XUIM_i + XTPM_i)}{n} \quad (37)$$

The value of the overall metrics shows that the explainable model is implemented successfully. The average acceptance scale to introduce the explainability for the end users in the web-based applications is based on the [30]. The following defined is the range of the metrics values:

$$\mathrm{Low}(<10),\ \mathrm{Medium}(10 to 20),\ \mathrm{High}(>20)$$

The highest value of the metric measured indicates that the model is implemented perfectly and achieves explainability in all aspects.

H. EXPLAINABLE DESIGN COMPLEXITY METRIC (EDCM)

Various components of the explainable model are implemented in the web-based application which defines the Explainable Design Quality Complexity Metric given as follows:

$$EDCM = \sum_{v=1,i_v=1}^{m,m_v} \alpha_v^{(i_v)} + \sum_{w=1,i_w=1}^{n,n_w} \beta_w^{(i_w)} \sum_{x=1,i_x=1}^{o,o_x} \delta_x^{(i_x)} + \sum_{y=1,i_y=1}^{p,p_y} \varepsilon_y^{(i_y)} + \sum_{z=1,i_z=1}^{q,q_z} \lambda_z^{(i_z)}$$

The unique value is assigned to each relation of edge as a weight value. The average design complexity of an explainable web-based application is calculated as

$$AverageDesignComplexity = \frac{\text{Total Number of Relations}}{\text{Total Number of Components}}$$

This metric defines the overall complexity of the web-based application by introducing the explainability and interaction of its various components.

IV. RESULTS AND DISCUSSIONS

The web-based multi-domain sentiment analysis application is considered as a case study to evaluate the design quality metrics of the explainable model defined in this work. In this web-based application for multidomain sentiment analysis, various explainable models are introduced to provide the detail of each module or complete web-based application at the user end. In the multidomain sentiment analysis, users interact with the application and give input as a sentence of a specific domain and get the output in the form of positive, negative, or neutral sentiment. The complete application provides the details of each feature to the user to enhance the fairness of data and user interest in the application. We consider the web-based application for the multidomain sentiment analysis which provides the explainability using various parameters illustrated in Figure 5.

Various numbers of users interact with the application and the defined parameters ensure the explainability of implementing numerous factors in the application. The factors include authentication, authorization, data availability, system failure, and information leakage. These factors provide the information to the respective component which further transfers to each related model. This information enables the explainable model, which helps enhance the system's

TABLE 2. End level parameters and their working in web-based application.

End Level Parameters	Details of Parameters
Encryption (EN)	Encode user data
Authentication (AC)	Validate user record
Authorization (AR)	Provide authority to access application
Uptime of the application (UT)	System is available
Downtime of the application (DT)	System is unavailable
Failure rate of application (FR)	Frequency of failures within a specific period
Availability (AV)	System is operational
Mean Time to Failure (MTTF)	Average time between current uptime and next failure
Mean Time between Failure (MTBF)	Average time between consecutive failures
Information Leakage (IL)	Sensitive information disclosed
Normalization (NM)	Measure the case sensitivity of the data
Cleaning (CL)	Remove low significance words
Term Frequency - Inverse Document Frequency (TF-IDF)	Most repeated terms in document
Transferability to Next Module Score (TNMS)	Performance of transitioning between previous and current module
Relevancy (RL)	Provide related information
Completeness (CT)	Covering all aspects of the information
Timeliness (TL)	Provide time-sensitive information
Language relevancy (LR)	User understandable language
Effectiveness of Visualization (EV)	Provide detail and clear visualization
Transparency Score (TC)	Provide transparency of the system
Complexity Score (CS)	Provide a simple interface to the user
Edges (E)	Number of edges in the control graph
Nodes (N)	Number of nodes in the control graph
Connected Components (CP)	Number of connected components in the control graph
Bias Detection Score (BDS)	Bias exist in system
User control Score (UCS)	System are controlled by user

performance and provides the user with a secure, transparent, interpretable, and interactive system. The system's explainability can be checked through these parameters. Moreover, these parameters can be incorporated according to the user's requirements.

To evaluate the defined metrics, we consider the different scenarios of the case study. Table 2 defines the low-level parameters and their working. These parameters help to implement the explainability in the overall application for the user and transfer data from the databases to the user interface. The user gets the details of each module in the web-based application through the implemented parameters which helps to deal with the transparent system.

A. EXPLAINABLE COMPLIANCE MODEL METRICS (XCMM)

The explainable compliance model metric is evaluated with three component scenarios of the web-based multi-domain sentiment analysis. Scenario 1 is presented in Figure 6 where the compliance model implements the trustworthiness through different parameters having encryption, authentication, and authorization. Incorporating trustworthiness, only authentic users are allowed to interact with the application which enhances the confidence of the users in the application. The transaction of the explainability model is shown in blue lines towards the compliance model for this condition, and these data read the explainability level of the model using (8) which defines the number of edges involved in this activity.

$$\begin{aligned} XCMM &= \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \beta_1^1 + \delta_1^1 + \varepsilon_1^1 + \lambda_1^1 1 \\ &\quad + 1 + 1 + 1 + 1 + 1 + 1 = 7 \text{ units} \end{aligned} \tag{38}$$

The implementation of the trustworthiness in compliance model takes 7 units calculated in (38). By implementing this scenario, we achieve trustworthiness in the compliance model of explainability.

In Scenario 2, as shown in Figure 7 blue and green highlighted edges towards the compliance model are used to implement trustworthiness with reliability in the compliance model. With the advent of reliability, users are not only confident but also satisfied by providing a stable and predictable outcome. The explainability level of the complete system to achieve trustworthiness and reliability in web-based applications is 14 units presented in (39).

$$\begin{aligned} XCMM &= \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4 + \alpha_1^5 + \alpha_1^6 + \alpha_1^7 + \alpha_1^8 + \alpha_1^9 \\ &\quad + \beta_1^1 + \beta_1^2 + \delta_1^1 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 \\ &\quad + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 14 \text{ units} \end{aligned} \tag{39}$$

Scenario 3 is presented in Figure 8 with blue, green, and orange edges towards the compliance model. In this scenario, the overall compliance model is implemented by incorporating trustworthiness, reliability, and privacy awareness. It enhances the system's ability to secure the user's data and inform them about its usage. Privacy awareness parameters along with trustworthiness and reliability provide trust, confidence, and stability, which ensure compliance with privacy regulations. All the parameters are implemented such as EN, AC, AR, UT, DT, MTTF, MTBF, FR, AR, and IL. The given components are merged to introduce the compliance model to achieve explainability in the web-based multi-domain sentiment analysis application.

The total explainability level to implement the compliance model in the explainable model is 16 units discussed in (40).

$$\begin{aligned} XCMM &= \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4 + \alpha_1^5 + \alpha_1^6 + \alpha_1^7 + \alpha_1^8 \\ &\quad + \alpha_1^9 + \alpha_1^{10} + \beta_1^1 + \beta_1^2 + \beta_1^3 + \delta_1^1 + \varepsilon_1^1 + \lambda_1^1 1 \\ &\quad + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &\quad + 1 + 1 + 1 + 1 = 16 \text{ units (No. of all edges)} \end{aligned} \tag{40}$$

The average explainability level of the compliance model in the explainable model for this case study is calculated to be (7 + 14 + 16) / 3 = (37) / 3 = 12 which is acceptable in this scenario.

B. EXPLAINABLE DATA-CENTRIC MODEL METRICS (XDCMM)

In the web-based multi-domain sentiment analysis application, data should be secured and understandable to the user. Here is the data related to the multidomain sentiment

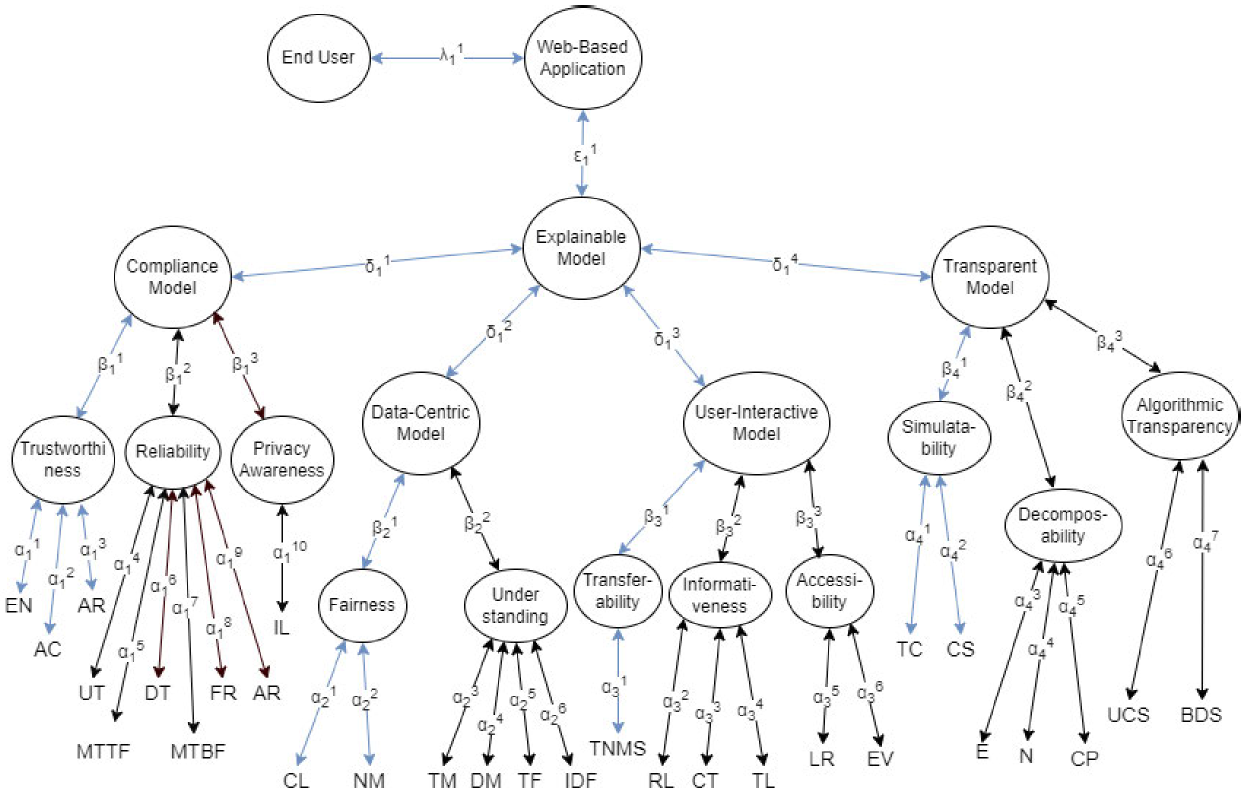

FIGURE 6. Scenario 1 for explainable model.

analysis from the social media platform. Two scenarios are implemented which help to evaluate the data-centric model to introduce explainability. In Figure 6 scenario 1, only one component is implemented in the data-centric model. The blue line towards the data-centric model with a fairness component is implemented in this scenario. The explainability level to achieve fairness is calculated in (41) which means the count of the total number of edges used in this path.

$$XDCMM = \alpha_2^1 + \alpha_2^2 + \beta_2^1 + \delta_1^2 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 + 1 = 6 \text{ units} \tag{41}$$

To introduce the fairness parameters in the data-centric model for explainable web-based applications, 6 edges are used which means it also has the complexity of 6 units.

Scenario 2 for the data-centric model, Figure 7 illustrates the blue and green edges towards the data-centric model that merge both parameters including fairness and understanding to get insight into the data for multi-domain sentiment analysis. The explainability to enhance the user's trust in the system, the complete detail of data is provided at the user-end which takes the explainability level is 11 units as mentioned in (42). In this scenario, the more explainability level predicts that the system is more explainable to the user as compared to scenario 1.

$$XDCMM = \alpha_2^1 + \alpha_2^2 + \alpha_2^3 + \alpha_2^4 + \alpha_2^5 + \alpha_2^6 + \beta_2^1 + \beta_2^2 + \delta_1^2 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 11 \text{ units} \tag{42}$$

The data-centric model provides explainability by incorporating two parameters which are fairness and understanding. Fairness provides equitable outcomes and insight to facilitate user comprehension, ensuring data acceptance and clarity to the users. Therefore, there is no need to implement this model in scenario 3. The average level of explainability to implement it is calculated as follows (6 + 11) / 2 = (17) / 2 = 8 units.

C. EXPLAINABLE USER-INTERACTIVE MODEL METRICS (XUIMM)

In the user-interactive model, the aim is to make the system explainable at the user interface. To introduce this explainability, we implement three parameters in this model. Therefore, the explainability level of the user-interactive model is measured with three scenarios. Figure 6 depicts scenario 1 where the blue edges are moved from the user-interactive model to the transferability factors. The total number of edges is the level of explainability for this parameter. Transferability refers to how well the components of each module interact with the others. It ensures the data in different modules is effectively navigated. The explainability level of transferability parameters is calculated by using (43) which is 5 units.

$$XUIMM = \alpha_3^1 + \beta_3^1 + \delta_1^3 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 = 5 \text{ units} \tag{43}$$

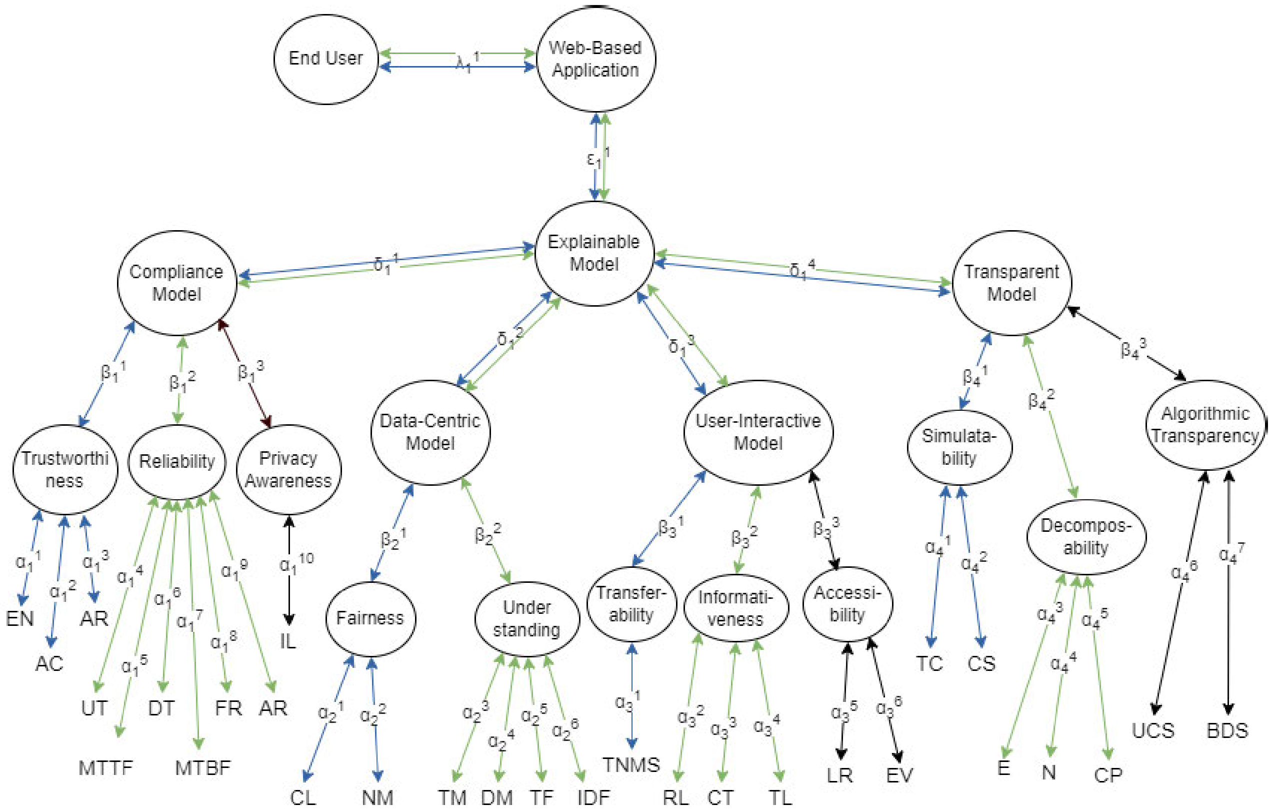

FIGURE 7. Scenario 2 for explainable model.

In scenario 2 for the user-interactive model, the transferability and informativeness parameters are implemented. In Figure 7, all the edges from the end user to the explainable model and then from the explainable model to the user-interactive model are involved. Through informative metrics, we calculated whether the data is relevant, clear, and achieves timeliness. This parameter helps the user to understand the depth and clarity of the information conveyed. The blue and green edges are used to calculate the explainability level of the defined parameters. The (44) describes the explainability level, which is 9 units to implement in this model.

$$XUIMM = \alpha_3^1 + \alpha_3^2 + \alpha_3^3 + \alpha_3^4 + \beta_3^1 + \beta_3^2 + \delta_1^3 + \varepsilon_1^1 + \lambda_1^1 1 \\ + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 9 \text{ units} \tag{44}$$

The explainability of the web-based application can be extended by incorporating one more parameter in the user-interactive model, which is accessibility. In scenario 3, we incorporate all three defined models, which enhance the explainability of the system. Figure 8 depicts the blue, green, and orange edges passed towards the user-interactive model. With these parameters, users can obtain and comprehend the application's explanations with a focus on user interface design, language clarity, and the availability of support resources. Therefore, we select all the edges of the model to calculate its explainability level, which is 12 units defined in (45).

$$XUIMM = \alpha_3^1 + \alpha_3^2 + \alpha_3^3 + \alpha_3^4 + \alpha_3^5 + \alpha_3^6 + \beta_3^1 + \beta_3^2 + \beta_3^3 \\ + \delta_1^3 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ + 1 + 1 + 1 + 1 + 1 = 12 \text{ units} \tag{45}$$

The average level of explainability to implement all the parameters is calculated with defined factors such as TNMS, RL, CT, TL, LR, and EV. All the scenarios are considered to get the average level of the user-interactive model which is $(5 + 9 + 12) / 3 = (26) / 3 = 8$ units.

D. EXPLAINABLE TRANSPARENT MODEL METRICS (XTPMM)

The transparent model is evaluated in three scenarios of the web-based multi-domain sentiment analysis to introduce the explainability in it. Scenario 1 is shown in Figure 6 where the blue edges are used to create the link between nodes. In this model, we count the number of edges that link to each factor of the simulatability parameter considered as an explainability level. Simulatability enhances the ability to predict the behavior of the functions that come under the simple and compact category of the functions that grasp the inputs to transform them into outputs without cognitive load. The (46) provides the explainability level which is 6 units.

$$XTPMM = \alpha_4^1 + \alpha_4^2 + \beta_4^1 + \delta_1^4 + \varepsilon_1^1 + \lambda_1^1 1 \\ + 1 + 1 + 1 + 1 + 1 = 6 \text{ units} \tag{46}$$

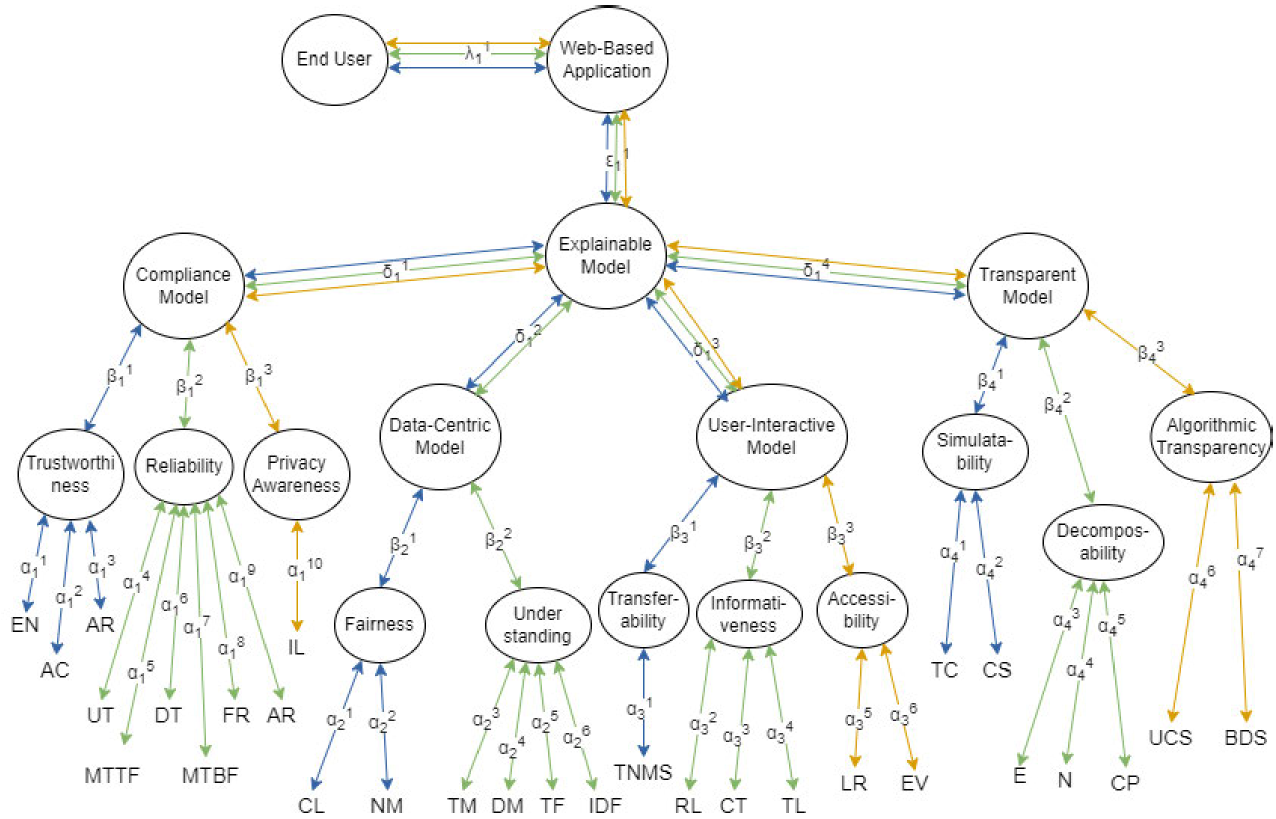

FIGURE 8. Scenario 3 for explainable model.

Figure 7 is used to implement scenario 2 in the transparent model. We consider two parameters, such as simulatability and decomposability to evaluate the explainable level of this model. Both parameters collectively allow users to analyze each function and their relationships to understand their contribution to the overall application. The blue and green edges towards the transparent model are used to calculate the level of explainability. The 10 units of explainability level introduced in this scenario are mentioned in (47).

$$\begin{aligned} XTPMM &= \alpha_4^1 + \alpha_4^2 + \alpha_4^3 + \alpha_4^4 + \alpha_4^5 + \beta_4^1 + \beta_4^2 + \delta_1^4 \\ &\quad + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &\quad + 1 = 10 \text{ units} \end{aligned} \tag{47}$$

The three parameters are defined in the transparent model including simulatability, decomposability, and algorithmic transparency. In scenario 3, as defined in Figure 8, we consider all these parameters to measure the explainability level of the model. The blue, green, and orange edges are calculated and move towards the transparent model. The major emphasis of the explainability level is the factors that are defined in this model such as TC, CS, E, N, CP, UCS, and BDS that measure the overall bias and accountability of the application data leading to more understandable and transparent application for the users at design level.

$$\begin{aligned} XTPMM &= \alpha_4^1 + \alpha_4^2 + \alpha_4^3 + \alpha_4^4 + \alpha_4^5 + \alpha_4^6 + \alpha_4^7 \\ &\quad + \beta_4^1 + \beta_4^2 + \beta_4^3 + \delta_1^4 + \varepsilon_1^1 + \lambda_1^1 1 + 1 + 1 + 1 \\ &\quad + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &\quad + 1 + 1 = 13 \text{ units} \end{aligned} \tag{48}$$

The total explainability level of the transparent model is 13 units mentioned in (48) the Average explainable level of the transparent model is calculated to be (6 + 10 + 13) /3 = (29) / 3 = 9, which is acceptable for this model.

E. EXPLAINABLE DESIGN COMPLEXITY METRIC (EXDCM)

The design complexity metrics determine the overall complexity of the web-based application to introduce the explainable model. Different scenarios are considered to measure the explainable level of complexity with various parameters of several models. Figure 6, Figure 7, and Figure 8 depict the complete scenarios to calculate complexity by providing details of each model. The design complexity of this web-based application is measured as an explainability level with with the IDMC metrics by evaluating the unit cost of each factor and parameters involved in the model.

$$\begin{aligned} EXDCM &= \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4 + \alpha_1^5 + \alpha_1^6 + \alpha_1^7 + \alpha_1^8 \\ &\quad + \alpha_1^9 + \alpha_1^{10} + \alpha_2^1 + \alpha_2^2 + \alpha_2^3 + \alpha_2^4 + \alpha_2^5 + \alpha_2^6 \\ &\quad + \alpha_3^1 + \alpha_3^2 + \alpha_3^3 + \alpha_3^4 + \alpha_3^5 + \alpha_3^6 + \alpha_4^1 + \alpha_4^2 \\ &\quad + \alpha_4^3 + \alpha_4^4 + \alpha_4^5 + \alpha_4^6 + \alpha_4^7 + \beta_1^1 + \beta_1^2 + \beta_1^3 \\ &\quad + \beta_2^1 + \beta_2^2 + \beta_3^1 + \beta_3^2 + \beta_3^3 + \beta_4^1 + \beta_4^2 + \beta_4^3 \\ &\quad + \delta_1^1 + \delta_1^2 + \delta_1^3 + \delta_1^4 + \varepsilon_1^1 + \lambda_1^1 \\ &= (29 + 11 + 4 + 1 + 1) = 46 \end{aligned} \tag{49}$$

TABLE 3. Explainable path of design for web-based applicationc.

Evaluation Scenario	Nodes	Edges	Explainable Path of Design	Weights of Paths
Scenario 1	Low-End	α	Max (α)	8
	Component Level			
	Basic Model Level	β	Max ($\sum \alpha_x^y$)	4
	Upper Model Level			
	User Level	δ	Max (β)	4
Scenario 2	Low-End	α	Max (α)	24
	Component Level			
	Basic Model Level	β	Max ($\sum \alpha_x^y$)	8
	Upper Model Level			
	User Level	δ	Max (β)	4
Scenario 3	Low-End	α	Max (α)	29
	Component Level			
	Basic Model Level	β	Max ($\sum \alpha_x^y$)	11
	Upper Model Level			
	User Level	δ	Max (β)	4

The average design complexity is 15 units in this case study of a web-based multi-domain sentiment analysis application which is calculated as (46/47) in (49). It is the perfect case as it is less than the total number of components. Therefore, our design-level explainable model is an extremely simple system for web-based applications.

F. DISCUSSION

The main aim of this proposed work was to provide the measurement criteria to incorporate the explainability at the design level of web-based applications. Existing work only quantifies the analyzability [31], authorization [32], and confidentiality [33] parameters at the design level [34]. For the quantitative assessment of the explainable model, we introduce the design quality metrics for the evaluation. We used the web-based application as a base and then implemented the explainable model as an abstract graph. The design models are defined to formally define the design quality metrics for web-based applications. Various metrics are defined, including Explainable Compliance Model Metrics, Explainable Data-Centric Model Metrics, Explainable User-Interactive Model Metrics, Explainable Transparent Model Metrics, and Explainable Design Complexity Metrics. The defined metrics are assessed using the case study of a web-based multi-domain sentiment analysis application. Different scenarios are implemented in each metric, which gives a variety of results. We compare the values of each scenario with the assigned range in this work.

Table 3 presents the calculated results of the overall scenario which are then compared with the realistic outcome. The results of the defined metrics are validated with the benchmark values that are taken from experts in web-based applications. Defined metrics give accurate values that are consistent with the predicted parameters. We can compare the design quality of different models for explainable web-based applications by using these results. The best model is the one that gives the higher explainability level in the web-based application which is the one from the defined metrics that have the maximum values.

V. CONCLUSION

This paper presented the evaluation of design quality metrics of an explainable model for web-based multi-domain sentiment analysis applications as a schema graph. The metrics are proposed for various explainable models including explainable compliance model metric, explainable data-centric model metric, explainable user-interface model metric, explainable transparent model metrics, and overall design metric of the explainable model. These metrics are mapped with the quality parameters of each model metric such as trustworthiness, reliability, privacy awareness, fairness, understanding, transferability, informativeness, accessibility, simulatability, decomposability, and algorithmic transparency. These quality parameters are further mapped with the encryption, authentication, authorization, uptime, downtime, failure rate, availability, mean time between failure, mean time to failure, information leakage, normalization, cleaning, term frequency-inverse document frequency, transferability, relevancy, completeness, timeliness, and effectiveness of the visualization. The proposed metrics are then evaluated on different case studies of the web-based multi-domain sentiment analysis application. Three different scenarios are implemented to compare the result, and the 3rd scenario gives a higher explainability level compared to the other scenarios. It covers all aspects to provide a clear and understandable design to the end users. Furthermore, the defined quality metrics are used to measure the design level explainability of the quality of explainable web-based applications. The work can be extended to evaluate real-time large data using the defined explainable model. These metrics were validated using the explainable AI models and compared the accuracy of the models and directly computed the output by comparing with metrics. These metrics can be used anywhere in applications to measure the explainability of the application at user-end.

DISCLOSURE STATEMENT

There are no relevant financial or non-financial competing interests of authors to report.

REFERENCES

[1] F. Gwinner, C. Tomitza, and A. Winkelmann, "Comparing expert systems and their explainability through similarity," *Decis. Support Syst.*, vol. 182, Jul. 2024, Art. no. 114248. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923624000812

[2] A. Vultureanu-Albisi and C. Badica, "Recommender systems: An explainable AI perspective," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6.

[3] H. S. Xavier, "The web unpacked: A quantitative analysis of global web usage," 2024, *arXiv:2404.17095*.

[4] A. Shah, "OODM: An object-oriented design methodology for development of web applications," in *Information Modeling for Internet Applications*. Hershey, PA, USA: IGI Global, 2003, pp. 189–229.

[5] I. J. Walker, "English requirements of an object-oriented design method," *English Softw. Eng. J.*, vol. 7, pp. 102–113, Mar. 1992.

[6] A. Farooq, S. Ahsan, and A. Shah, "Engineering semantic web applications by using object-oriented paradigm," 2010, *arXiv:1006.4562*.

[7] M. U. Ghani, M. Zafar, M. J. Arshad, K. H. Asif, and A. A. Shah, "Reverse object oriented design methodology (R-OODM)," *J. Fac. Eng. Technol.*, vol. 21, no. 2, pp. 60–81, 2014.

[8] S. Arshad, "S-OODM: A secure object oriented design methodology," *Pakistan J. Sci.*, vol. 62, no. 1, pp. 51–54, Mar. 2010.
[9] H. T. Phan, N. T. Nguyen, V. C. Tran, and D. Hwang, "A sentiment analysis method of objects by integrating sentiments from tweets," *J. Intell. Fuzzy Syst.*, vol. 37, no. 6, pp. 7251–7263, Dec. 2019.
[10] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai-explainable artificial intelligence," *Sci. Robot.*, vol. 4, no. 37, 2019, Art. no. eaay7120. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aay7120
[11] 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.*, vols. 7–8, Oct. 2021, Art. no. 100017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266665962100010X
[12] 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.
[13] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable AI (XAI): Core ideas, techniques, and solutions," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–33, Jan. 2023, doi: 10.1145/3561048.
[14] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing design practices for explainable AI user experiences," in *Proc. CHI Conf. Hum. Factors Comput. Syst.* New York, NY, USA: ACM, Apr. 2020, pp. 1–15, doi: 10.1145/3313831.3376590.
[15] M. Aniche, J. Yoder, and F. Kon, "Current challenges in practical object-oriented software design," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., New Ideas Emerg. Results (ICSE-NIER)*, May 2019, pp. 113–116.
[16] E. J. Aas, "Design quality and design efficiency; definitions, metrics and relevant design experiences," in *Proc. IEEE 1st Int. Symp. Quality Electron. Design*, Jul. 2000, pp. 389–394.
[17] Y. Pan, B. Wu, H. Zheng, Y. Zong, and C. Wang, "The application of social media sentiment analysis based on natural language processing to charity," in *Proc. The 11th Int. Sci. Practical Conf.* Boston, MA, USA: International Science Group, Mar. 22, 2024, p. 216.
[18] R. Marinescu, "Measurement and quality in object-oriented design," in *Proc. 21st IEEE Int. Conf. Softw. Maintenance (ICSM)*, Aug. 2005, pp. 701–704.
[19] W.-Y. Liang and P. O'Grady, "Design with objects: An approach to object-oriented design," *Comput.-Aided Des.*, vol. 30, no. 12, pp. 943–956, Oct. 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010448598000505
[20] M. Kiewkanya, N. Jindasawat, and P. Muenchaisri, "A methodology for constructing maintainability model of object-oriented design," in *Proc. 4th Int. Conf. Quality Softw., QSIC*, vol. 6, 2004, pp. 206–213.
[21] S. P. Kadam and S. Joshi, "Secure by design approach to improve security of object oriented software," in *Proc. 2nd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2015, pp. 24–30.
[22] K. Jangra and K. Dua, "Health care management information systems with object oriented methodology," *Int. J. Inf. Technol.*, vol. 5, no. 1, pp. 234–239, 2012.
[23] S. Mohapatra, K. Paul, and A. Roy, "Object-oriented modeling of cloud healthcare system through connected environment," in *Applications of Internet of Things*, J. K. Mandal, S. Mukhopadhyay, and A. Roy, Eds., Singapore: Springer, 1007, pp. 151–164.
[24] I. Nwokoro, O. Okonkwo, R. U. Alo, and A. N. Nwatu, "Application of object-oriented analysis and design methodology in modeling of university administrative information system," *Academia Lett.*, pp. 1–11, Nov. 2021, Art. no. 3853, doi: 10.20935/AL3853.
[25] E. Arulprakash and A. Martin, "An object-oriented neural representation and its implication towards explainable AI," *Int. J. Inf. Technol.*, vol. 16, no. 3, pp. 1303–1318, Mar. 2024.
[26] P. Geyer, M. M. Singh, and X. Chen, "Explainable AI for engineering design: A unified approach of systems engineering and component- based deep learning demonstrated by energy- efficient building design," 2021, *arXiv:2108.13836*.
[27] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103
[28] T. D. Huynh, N. Tsakalakis, A. Helal, S. Stalla-Bourdillon, and L. Moreau, "A methodology and software architecture to support explainability-by-design," 2022, *arXiv:2206.06251*.
[29] J. H.-W. Hsiao, H. H. T. Ngai, L. Qiu, Y. Yang, and C. C. Cao, "Roadmap of designing cognitive metrics for explainable artificial intelligence (XAI)," 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, Dec. 1989, doi: 10.1145/76380.76382.
[31] S. A. Khan and R. A. Khan, "Analyzability quantification model of object oriented design," *Proc. Technol.*, vol. 4, pp. 536–542, May 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017312003647
[32] S. A. Khan and R. A. Khan, "Securing object oriented design: A complexity perspective," *Int. J. Comput. Appl.*, vol. 8, no. 13, pp. 8–12, Oct. 2010.
[33] S. A. Khan and R. A. Khan, "Confidentiality quantification model at design phase," *Int. J. Inf. Educ. Technol.*, vol. 2, no. 5, pp. 535–537, Oct. 2012.
[34] S. A. Khan, M. Nadeem, A. Agrawal, R. A. Khan, and R. Kumar, "Quantitative analysis of software security through fuzzy PROMETHEE-II methodology: A design perspective," *Int. J. Modern Educ. Comput. Sci.*, vol. 13, no. 6, pp. 30–41, Dec. 2021.

ABQA JAVED received the B.Sc. degree (Hons.) from the Islamia University of Bahawalpur, and the master's degree from the University of Engineering and Technology (UET) at Lahore, Pakistan.

She is currently a Lecturer with the UET Lahore. Her research interests include machine learning, data mining, recommendation systems, explainability, and software metrics.

MUHAMMAD SHOAIB received the Ph.D. degree in computer science from the University of Engineering and Technology, Lahore, Pakistan, in 2006.

He also received a Postdoc from the Florida Atlantic University (FAU), USA, in 2009. He is currently serving as a Professor at the Computer Science Department and Dean of the Faculty of Electrical Engineering, at the University of Engineering and Technology. His research interests include advanced databases, object-oriented design and analysis, design quality metrics, and information retrieval systems.

ABDUL JALEEL received the B.S., M.S., and Ph.D. degrees in computer science and engineering from the University of Engineering and Technology at Lahore, Pakistan, in 2006, 2010, and 2019, respectively.

He is currently an Associate Professor with the Rachna College of Engineering Technology, University of Engineering and Technology at Lahore. His research interests include the development of self-managing software applications, the health IoT, autonomic computing, and software quality measurement metrics.