Explaining Sentiments: Improving Explainability in Sentiment Analysis Using Local Interpretable Model-Agnostic Explanations and Counterfactual Explanations

Xin Wang, *Member, IEEE*, Jianhui Lyu, *Member, IEEE*, J. Dinesh Peter, *Member, IEEE*, Byung-Gyu Kim, *Senior Member, IEEE*, B.D. Parameshachari, *Senior Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Wei Wei, *Senior Member, IEEE*

***Abstract*—Sentiment analysis of social media platforms is crucial for extracting actionable insights from unstructured textual data. However, modern sentiment analysis models using deep learning lack explainability, acting as black box and limiting trust. This study focuses on improving the explainability of sentiment analysis models of social media platforms by leveraging explainable artificial intelligence (XAI). We propose a novel explainable sentiment analysis (XSA) framework incorporating intrinsic and posthoc XAI methods, i.e., local interpretable model-agnostic explanations (LIME) and counterfactual explanations. Specifically, to solve the problem of lack of local fidelity and stability in interpretations caused by the LIME random perturbation sampling method, a new model-independent interpretation method is proposed, which uses the isometric mapping virtual sample generation method based on manifold learning instead of LIMEs random perturbation sampling method to generate samples. Additionally, a generative link tree is presented to create counterfactual explanations that maintain strong data fidelity, which constructs counterfactual narratives by leveraging examples from the training data, employing a divide-and-conquer strategy combined with local greedy. Experiments conducted on social media datasets from Twitter, YouTube comments, Yelp, and Amazon demonstrate XSAs ability to provide local aspect-level explanations while maintaining sentiment analysis performance. Analyses reveal improved model explainability and enhanced user trust, demonstrating XAIs potential in sentiment analysis of social media platforms. The proposed XSA framework provides a valuable direction for developing transparent and trustworthy sentiment analysis models for social media platforms.**

***Index Terms*—Explainable artificial intelligence (XAI), explainability, local interpretable model-agnostic explanations (LIME), sentiment analysis (SA).**

I. Introduction

SENTIMENT analysis, or opinion mining, is the computational study and extraction of subjective information such as opinions, emotions, attitudes, or sentiments from textual data [1]. The proliferation of social media platforms such as Twitter, Facebook, and forums has led to exponential growth in user-generated textual content containing subjective opinions on diverse topics [2], [3]. Sentiment analysis of such social media text has thus emerged as a critical technique to derive actionable insights into public opinion, attitudes, trends, and behavioral psychology [4]. Some significant applications of sentiment analysis focused on social media data include [5], [6], [7].

In recognizing the exponential growth in user-generated content on various social media platforms, it becomes increasingly evident that the volume and diversity of this data significantly amplify the complexity of sentiment analysis tasks. The myriad of sources, ranging from Twitter to Instagram, not only introduces a variety of content styles and formats but also necessitates scalable solutions capable of efficiently processing large datasets. Furthermore, the evolving language on these platforms, characterized by slang, emojis, and informal expressions, poses unique challenges. This necessitates advanced sentiment analysis models that can adeptly navigate and interpret social media posts' nuanced and often ambiguous context.

Social media sentiment analysis represents a significant advancement in gauging public opinion, harnessing the expansive

Received 10 November 2023; revised 22 January 2024 and 4 November 2024; accepted 25 December 2024. Date of publication 8 April 2025; date of current version 2 June 2025. This work was supported by the National Natural Science Foundation of China under Grant 62202247. *(Corresponding author: Jianhui Lyu).*

Xin Wang is with the Northeastern University, Shenyang 110819, China (e-mail: dnsy_heinrich@neueet.com).

Jianhui Lyu is with The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121001, China (e-mail: lvjianhui2012@163.com).

J. Dinesh Peter is with Karunya Institute of Technology and Sciences, Tamil Nadu 641114, India (e-mail: dineshpeterson@gmail.com).

Byung-Gyu Kim is with Sookmyung Women's University, Seoul 04310, Republic of Korea (e-mail: bg.kim@sookmyung.ac.kr).

B.D. Parameshachari is with Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka 560064, India (e-mail: paramesh@nmit.ac.in).

Keqin Li is with The State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Wei Wei is with Xi'an University of Technology, Xi'an 710048, China, and also with the University of Wollongong, Sydney, NSW 2522, Australia (e-mail: weiwei@xaut.edu.cn).

Digital Object Identifier 10.1109/TCSS.2025.3531718

power of the internet to capture a wide array of sentiments that traditional methods might miss. Unlike surveys or interviews, which can be limited in reach and often come with significant delays and costs, sentiment analysis taps into the spontaneous and candid expressions of millions of users, providing a treasure trove of data that reflects a broad spectrum of public opinion [8], [9], [10], [11]. The instantaneous nature of social media allows for the observation and analysis of real-time data, allowing organizations to react promptly to public sentiment trends. This immediacy is crucial when a timely understanding of public opinion can guide crucial decisions or strategies. Moreover, social media platforms serve as a natural environment where users feel free to express their opinions without the influence of a surveyor or interviewer, potentially leading to more honest and raw reflections of their sentiments.

For instance, a significant brand's sentiment analysis algorithm misinterpreted ironic and sarcastic comments on social media as positive feedback, leading to misguided marketing strategies. Another example pertains to political sentiment analysis, where algorithms are needed to discern the contextual meaning behind specific colloquial phrases used in political discussions, resulting in an inaccurate assessment of public opinion. These instances exemplify the challenges in deciphering online communication's nuanced and often informal nature and demonstrate the potential repercussions of such misinterpretations in practical applications.

Analyzing sentiment on social media is a complex task due to online communication's unique and unstructured nature [12]. The content on these platforms needs to be more consistent with informal language, including spelling mistakes, slang, and creative use of emoticons, which can obscure the intended sentiment. Moreover, sentiment is not just expressed through text; it permeates multimedia content such as images, videos, and audio, necessitating sophisticated analysis tools capable of interpreting various signals and cues across various formats [13], [14]. This multifaceted nature of social media demands advanced algorithms and nuanced approaches to effectively capture and interpret the broad spectrum of human emotions and opinions expressed online.

To overcome these challenges, cutting-edge techniques using deep neural networks such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer networks have driven significant advances in social media sentiment analysis [15]. Models such as bidirectional encoder representations from transformers (BERT) and robustly optimized BERT pretraining approach (RoBERTa) pretrained on prominent social media text have achieved state-of-the-art performance across many sentiment analysis benchmarks. However, a significant limitation of such complex neural models is their need for more explainability. They behave like black boxes with opaque predictions to end users due to their inscrutable internal workings. For example, a healthcare application with a deep learning model for predicting patient outcomes made accurate predictions. However, its lack of transparency and explainability led to mistrust and reluctance among medical professionals to adopt the technology. Another instance involves a financial services firm where a nonexplainable deep learning model used for credit scoring was inadvertently found to reinforce discriminatory biases.

To tackle these issues, explainability has emerged as a critical requirement to build transparent and trustworthy sentiment analysis systems. Generating explanations that justify the rationale behind predictions is crucial for user acceptance and model validation. This requires explaining how the model generates sentiment predictions from social media text inputs. Explainable artificial intelligence (XAI) offers various techniques to uncover these explanations by explaining the reasoning behind model predictions [16].

This article focuses on leveraging diverse XAI methods to improve explainability in sentiment analysis models, specifically focused on the domain of social media text. The main contributions of this article can be summarized as follows.

1) *Explainable Sentiment Analysis Framework* (*XSA*)*:* We introduce an XSA framework that integrates intrinsic and posthoc XAI techniques, which aims to enhance the explainability of deep learning models for sentiment analysis, addressing the "black box" nature of current models and bolstering user trust through transparency.
2) *Isometric Mapping Virtual Sample Generation—local interpretable model-agnostic explanations* (*imVSG-LIME*)*:* To address the local fidelity and stability issues of LIME's random perturbation sampling, we propose a novel model-independent interpretation method. The imVSG uses manifold learning to generate virtual samples, providing a more reliable and stable basis for model explanations.
3) *Generative Link Tree for Counterfactual Explanations:* We present the generative link tree method to produce high-fidelity counterfactual explanations. By leveraging a divide-and-conquer strategy with a local greedy approach, the generative link tree generates explanations that closely reflect the training data, offering a more nuanced understanding of model decisions.

The rest of this article is structured as follows: Section II discusses the related works of this study, Section III outlines the XSA framework, Section IV presents the experiments, and Section V shows the conclusion.

II. Related Works

A. Sentiment Analysis

Early sentiment analysis models relied heavily on lexicon-based approaches for analyzing and understanding text sentiment. These models utilized sentiment lexicons, one prominent example being SentiWordNet [17], to determine the sentiment of individual words or phrases. In lexicon-based approaches, each word in a given text is assigned a sentiment score based on its presence in the sentiment lexicon. The sentiment scores of the identified lexicon words directly contribute to the predictions, which means that the sentiment analysis model can explicitly link its predictions to the sentiment scores assigned to the words in the sentiment lexicon. For example, suppose a lexicon-based sentiment analysis model determines that a text has a positive sentiment. In that case, it can trace this prediction back to words with positive sentiment scores in the sentiment

lexicon. This transparency allows users to understand why a particular sentiment was assigned to a given text, as they can see which words contributed to that sentiment. Nazir et al. [18] highlighted the difficulties associated with pinpointing various aspects and their corresponding sentiments, establishing connections between aspects, interactions, dependencies, and the contextual–semantic interplay among diverse data entities to enhance the precision of sentiment analysis, as well as forecasting the changing nature of sentiment over time. However, such approaches had limited context-handling capabilities.

Historically, lexicon-based approaches were foundational in sentiment analysis, primarily relying on a predetermined list of words with associated sentiment scores, by purely lexicon-based models in today's dynamic social media environment, where slang, idioms, and evolving language use can outpace static lexicons.

Modern sentiment models overcome this using machine learning such as support vector machine (SVM) and naive Bayes [5], [11]. Deep learning models such as CNN, RNN, and transformer networks further enhance context learning for sentiment analysis [19], [20], [21], [22]. However, complex non-linear mappings in these data-driven models reduce explainability. Liang et al. [23] introduced a graph convolutional network that utilized SenticNet to exploit affective dependencies within sentences based on specific aspects. Li et al. [24] presented a party-ignorant framework named bidirectional emotional recurrent unit, which was fast, compact, and parameter-efficient, designed for conversational sentiment analysis, called BiERU. Basiri et al. [25] proposed an attention-based bidirectional CNN–RNN deep model for sentiment analysis, named ABCDM. Yang et al. [26] introduced a novel sentiment analysis model that integrated sentiment lexicon, combining CNN with an attention-based bidirectional gated recurrent unit.

B. Explainable AI

Some studies try to enhance interpretability in deep sentiment models using techniques such as attention, sentence concatenation explanations, and concept-level explanations [16]. However, most focus only on intrinsic explainability within the model architecture itself. Posthoc explainability using XAI remains relatively unexplored for sentiment analysis.

The growing focus on explainable AI has led to the development of many XAI methods that allow explaining black-box model predictions. Techniques such as LIME and SHAP can link predictions to important input features [27], [28]. Counterfactual explanations [29] provide contrastive "what-if" explanations by perturbing inputs. Such posthoc XAI techniques have shown promise in improving explainability for NLP tasks. However, their applications in sentiment analysis still need to be improved.

Some recent works have tried to utilize XAI for aspects of sentiment analysis. Jiarpakdee et al. [30] improved LIME with hyperparameter optimization (LIME-HPO). However, there are potential limitations in terms of computational efficiency when applied to large-scale datasets. Lovera et al. [31] introduced a novel combined methodology that utilized knowledge graphs alongside deep learning methods to determine whether brief texts, such as Twitter messages, carry a positive or negative sentiment called KGDL-SA. Nevertheless, it depends on the quality and comprehensiveness of the knowledge graph used. Jain et al. [32] used an aware dictionary for sentiment reasoning (VADER), and LIME was used to provide in-depth insight into the predictions (VADER-LIME). Nevertheless, it has potential challenges in handling sarcasm and indirect expressions. Li [33] used SHAP to interpret extreme gradient boosting (XGBoost) as an example to demonstrate how to extract spatial effects from machine learning models (SHAP-XGBoost). However, its complexity and the potentially steep learning curve for practitioners are also the challenges. Huang et al. [34] presented an innovative model called AEC-LSTM, designed for detecting sentiment in text. This model seeks to enhance the capabilities of the traditional LSTM network by incorporating aspects of emotional intelligence and an attention mechanism.

In summary, while recent sentiment analysis models using deep learning have achieved high predictive performance, they lack inherent explainability due to their black box nature, which limits their transparency, adoption, and fairness, especially for critical applications such as social media platform analysis. Although some works have tried to incorporate intrinsic explainability in model architectures, posthoc explainability using diverse XAI techniques still needs to be explored for sentiment analysis, presenting an open research gap. Initial efforts have tried applying specific XAI methods such as LIME or SHAP in a restricted context. However, a dedicated focus on harnessing the full potential of multifaceted XAI to improve explainability in sentiment analysis holistically is needed.

III. Explainable Sentiment Analysis Framework

This section elucidates the workings of the XSA framework comprehensively. We provide detailed formulations and descriptions for each module of the XSA architecture. The XSA framework addresses the crucial tradeoff between model explainability and performance through a multifaceted approach. At its core, the framework employs a dual-model strategy, utilizing a high-performance "black-box" model for primary sentiment predictions alongside simpler, interpretable models for generating explanations. This allows the framework to maintain high accuracy while still providing explainable insights. By leveraging posthoc explanation techniques such as LIME and SHAP, XSA can offer explanations without modifying the underlying high-performance model, thus preserving its accuracy. The framework dynamically adjusts the complexity of explanations based on the confidence level of predictions, providing simpler explanations for high-confidence cases and more detailed ones for borderline predictions. To optimize the explanation generation process, XSA incorporates metrics that evaluate the fidelity of explanations to the original model and their comprehensibility to users. When fine-tuning models for specific platforms or domains, the framework employs regularization techniques that encourage sparsity and interpretability in the model's internal representations, striking a balance between performance and explainability. An ensemble approach combines predictions from complex and simple models, weighted by their respective performance and explainability

scores, helping maintain high accuracy while increasing overall model interpretability. The framework also implements a continuous evaluation process, monitoring user interactions with explanations to refine the explanation generation process over time. Finally, XSAs modular architecture allows for easy updating of individual components, enabling the incorporation of state-of-the-art models and explanation techniques as they become available. Through this comprehensive strategy, the XSA framework strives to provide meaningful explanations while maintaining high accuracy in sentiment analysis across various social media platforms.

A. Input Preprocessing

The raw textual data must be converted into a suitable numerical representation before machine learning algorithms can analyze it for sentiment prediction. This conversion is achieved through two key stages: tokenization and embedding.

In tokenization, the continuous sequence of characters in the textual data is split into smaller chunks called tokens. Tokens can be words, phrases, punctuation marks, or any meaningful linguistic unit. Tokenization transforms the free-flowing text into a discrete sequence of such tokens.

It is implemented by treating delimiters such as whitespace, commas, and periods as split points to break the text into tokens. For example, consider the sample sentence: "The food here is delicious!" This sentence will be split into the following tokens by treating whitespace and punctuation as delimiters: ["The," "food," "here," "is," "absolutely," "delicious," "!"]

Different tokenization schemes can be employed depending on the language of the text. For English text, punctuation marks and whitespaces are commonly used as delimiters for splitting into tokens. Using punctuation/whitespace delimited tokenization, the raw input text is converted into tokens. These tokens are then translated to numerical embedding vectors. More advanced tokenization techniques can handle aspects such as contractions and abbreviations based on linguistic rules specific to the language. Further details are provided comparing different tokenization schemes:

1) *Word Tokenization:* Simple tokenization based only on whitespaces and punctuation.
2) *Subword Tokenization:* Splits words into subwords using morphology, maintaining meaning.
3) *Character Tokenization:* Breaks text into individual characters as tokens.
4) *Linguistic Tokenization:* Uses linguistic grammar rules and dictionaries for context-aware token splitting, especially for languages such as Chinese.

While punctuation/whitespace tokenizes well for informal social media text, linguistic rules provide robustness for complex grammar and spellings. Subword encoding gives a midground, preventing out-of-vocabulary terms. The optimal scheme is task and language-dependent

$$x = (x_1, x_2, ..., x_n). \tag{1}$$

where n is the total number of tokens extracted from the input text, and x_i denotes an individual token.

Once tokenization splits the text into discrete tokens, each token x_i needs to be converted into a numerical vector representation v_i before it can be processed by machine learning algorithms. This numeric encoding of tokens is referred to as embedding.

The XSA framework could be extended with several specialized techniques to address common special cases in social media text. A preprocessing step using a dedicated model trained on labeled sarcastic tweets could be incorporated into the sentiment analysis pipeline for sarcasm detection. The sarcasm probability could then be used as an additional feature or to adjust sentiment scores. Emoji handling could be improved through emoji-to-text conversion using existing dictionaries or by training special embedding vectors for emojis to capture their sentiment connotations. These emoji embeddings could then be integrated with word embeddings in the input representation. To tackle multilingual mixing, the framework could employ language detection algorithms to identify primary and secondary languages in each input text, then utilize language-specific pretrained embeddings or multilingual models for processing mixed-language inputs. Context-aware processing could be enhanced by implementing attention mechanisms capable of capturing long-range dependencies, which is crucial for understanding sarcasm and mixed-language expressions. Data augmentation techniques could also generate synthetic examples of sarcastic, emoji-rich, and multilingual texts, improving model robustness during training. By implementing these extensions, the XSA framework would be better equipped to handle the complexities of social media text, potentially enhancing both sentiment analysis accuracy and the quality of generated explanations.

Embedding maps each token x_i to a dense vector $v_i \in \mathbb{R}^d$ where d represents the embedding dimension. The numerical vector v_i encodes semantic and syntactic properties of the token within the text.

The embedding size d encodes semantic complexity, with higher values capturing finer relationships at the cost of higher dimensionality. This dimensionality impacts model complexity—larger d leads to more parameters and computations. Typical values range from 100 to 1000, depending on dataset size and language complexity. Optimization is necessary to balance representational power and efficiency.

Various embedding techniques can be utilized such as word2-vec, global vectors for word representation, and BERT [35]. Each technique encodes information about the token into the d-dimensional vector differently based on its approach. The choice of embedding method depends on factors such as size of labeled training data available and complexity of the sentiment prediction model.

Applying embedding on each token generates the numeric sequence

$$x_{\text{emb}} = (v_1, v_2, ..., v_n) \tag{2}$$

where $v_i \in \mathbb{R}^d$ is the embedded vector corresponding to token x_i. This numerically represented sequence x_{emb} serves as the input to the sentiment prediction model.

The raw text is preprocessed into a vectorized sequence x_{emb} through tokenization which extracts discrete tokens, followed

by embedding which encodes the tokens numerically. This numeric representation x_{emb} can then be effectively processed by machine learning algorithms for sentiment analysis. The preprocessing steps transform the raw textual data into a representation suitable for the prediction model.

B. Sentiment Prediction Model

The sentiment prediction model is the core component of the XSA framework responsible for analyzing the textual input and generating the predicted sentiment label, which can be represented as follows:

$$f_{\text{sentiment}} : x_{\text{emb}} \rightarrow \hat{y} \tag{3}$$

where $\hat{y}$ is the sentiment label predicted by the model for the given text.

The model $f_{\text{sentiment}}$ can be instantiated using any suitable supervised machine learning or deep learning architecture. Some commonly used options include logistic regression, naive Bayes, SVM, CNN, recurrent networks such as long short-term memory (LSTM) network, and transformer networks such as BERT [36].

All these models need to be trained in a supervised manner on a dataset containing text examples labeled with their associated sentiment. This training dataset can be denoted as follows:

$$\mathcal{D} = (x_i, y_i)_{i=1}^{N} \tag{4}$$

where $\mathcal{D}$ represents the entire training dataset comprising N examples, x_i refers to the ith textual input, and y_i is the corresponding sentiment label for that text sample.

The model is trained by minimizing a loss function $\mathcal{L}$ that compares the predicted sentiment $\hat{y}_i$ with the ground truth sentiment label y_i for each training example. Cross-entropy loss is commonly used as the objective function for sentiment classification models. The loss is optimized over multiple iterations to learn the optimal parameters θ^* of the model

$$\theta^* = \arg\min \theta \sum (x_i, y_i) \in \mathcal{D}\mathcal{L}(f_{\text{sentiment}}(x_{\text{i}}; \theta), y_i). \tag{5}$$

This results in an optimized sentiment prediction model $f_{\text{sentiment}}(x_i; \theta)$ that can be applied on new unlabeled text inputs to predict their most likely sentiment label $\hat{y}$.

C. Intrinsic Explainability Models

The XSA framework employs simple yet intrinsically interpretable machine learning models to provide basic explanations for the predictions made by the sentiment model. These glass-box models include:

1) Linear Models: Linear classifiers such as logistic regression can be used to explain the predicted sentiment through importance weights assigned to input tokens based on the learned model parameters. Specifically, the weight matrix W learnt by logistic regression indicates the influence of each input token toward predicting a particular sentiment class.

For an input $x_{\text{emb}} \in \mathbb{R}^{n \times d}$, the prediction is made as follows:

$$\hat{y} = \text{softmax}(Wx_{\text{emb}} + b) \tag{6}$$

where $W \in \mathbb{R}^{C \times d}$ is the weight matrix with C being the number of sentiment classes, and $b \in \mathbb{R}^{C}$ is the bias vector.

By examining the magnitudes of the weights in W, users can gain insights into how the model works and identify tokens that are more influential in determining the positive or negative sentiment. For instance, a positive weight could signify the token contributes to a positive sentiment, while a negative weight suggests the token contributes more to a negative sentiment prediction.

Thus, linear models provide basic explainability by revealing input tokens deemed important by the model through the learned weight parameters. However, they tend to have lower predictive accuracy for sentiment analysis than nonlinear models.

2) Decision Trees: Decision trees work by recursively splitting the textual input space based on learned decision rules at internal nodes which finally lead to sentiment classification outcomes at the leaf nodes. Starting from the root node, the input text is assessed against the split rules at each internal node to traverse down a path until a leaf node is reached indicating the predicted sentiment class.

For instance, a sample decision tree for sentiment analysis could have rules such as

If word_count $<$ 15 goto Node 1 else Node 2

…

Node 1: Positive sentiment

Node 2: Negative sentiment

By tracing the path of triggered rules from root to leaf, intrinsic explanations for the predicted sentiment label can be obtained based on the input textual attributes present in the split rules along the path. However, as decision trees get larger in depth, interpreting them can become more difficult for users.

3) Rule-Based Models: Rule-based models explicitly model conditional rules to map input text to sentiment predictions. The rules have the following structure.

If {condition(s) on input tokens} Then {sentiment prediction}

Some sample rules are

If {POS (terrible) == ADJ} Then {Negative Sentiment}

If {DEP (food, terrible) == amod} Then {Negative Sentiment}

If {POS (amazing) == ADJ} and {DEP (ambiance, amazing) == amod} Then {Positive Sentiment}

Here POS and DEP refer to parts-of-speech and dependency relations between tokens. The conditions check for insightful textual attributes while the prediction consequent assigns a sentiment label.

These simpler models contribute significantly to the XSA framework's sentiment analysis capabilities. They provide inherently interpretable predictions, making the decision-making process more transparent to users. By serving as baselines, they help quantify performance gains of more advanced techniques. Linear models and decision trees can highlight influential input features, providing initial insights into model reasoning. Rule-based models have the advantage of encoding domain knowledge, improving robustness for certain linguistic nuances that statistical models might miss. By combining predictions from these simpler models with more complex ones, XSA generates multi-faceted explanations that may capture different aspects of expressed sentiment. This approach allows the framework to leverage the strengths of both simple and complex models, enhancing overall explainability without sacrificing performance.

D. Posthoc Explainability Modules: Proposed imVSG-LIME Model

The posthoc modules in XSA leverage advanced techniques to provide explanations by treating the sentiment model as a black box. The isometric mapping (Isomap) imVSG represents a virtual sample generation technique that is grounded in feature representation principles, which leverages the manifold learning technique of isometric mapping to reduce the dimensionality of the dataset [37]. Subsequently, interpolation techniques along with the utilization of the extreme learning machine [38] are employed to create virtual samples. The innovative method has the capability to produce locally robust and densely virtual samples. Building upon this concept, imVSG is integrated into the LIME framework as a replacement for the random perturbation sampling method for sample generation. Additionally, hierarchical clustering is applied to amalgamate and select representative samples for training the explanatory model.

The Isomap virtual sample generation technique in imVSG-LIME is implemented through a series of steps designed to preserve the intrinsic geometry of the data. Initially, Isomap is applied to reduce the dimensionality of the input data while maintaining its underlying structure. This process involves constructing a k-nearest neighbor graph in the reduced dimensional space, representing the data's local geometry. The shortest path distances on this graph are then computed, approximating the geodesic distances on the manifold. Classical multidimensional scaling is applied to this geodesic distance matrix, resulting in a low-dimensional embedding that preserves these distances. New virtual samples are generated within this embedded space by interpolating between existing data points, ensuring they lie on or close to the learned manifold. Finally, these generated samples are mapped back to the original feature space using techniques such as the Nystrom or neural network-based approaches. This comprehensive process ensures that the generated virtual samples maintain the intrinsic structure of the original data, leading to more reliable local explanations in the LIME framework.

The primary objective of LIME involves training a straightforward and interpretable model within the vicinity of the target instance for the purpose of explaining a specific prediction [39]. The explanation is derived by analyzing the coefficients within the explanatory model. To facilitate the training of the explanatory model, it is necessary to generate a batch of simulated data within the proximity of the instance being explained. LIME employs a random perturbation sampling technique to generate simulated data, but this method has certain drawbacks. First, the samples generated using random perturbation tend to be widely dispersed, and some may deviate from the original data distribution. This deviation has a significant negative impact on the local fidelity of the explanatory model. Reduced local fidelity implies that the explanation method lacks reliability. Then, due to the inherent randomness of this approach, repeated experiments conducted under identical conditions for the same instance being explained will yield different sets of samples. This variability introduces instability into LIMEs explanations. An unstable interpretation results in explanations that need more plausibility.

For a given black-box model f and an instance x to be explained, imVSG-LIME produces explanations through the following steps:

1) Neighbor Selection: The imVSG generation model requires a certain number of base samples as input, while the samples generated by the model should be as dense as possible. Therefore, m samples that are closest to the instances to be explained are selected from the training set by calculating the Euclidean distance, as follows:

$$\text{Neighbors} = \arg\min_{m} \sqrt{\sum_{i=1}^{n} (x_i - x_{\text{train}_i})^2} \tag{7}$$

where x_i is the featur of the instance x, x_{train_i} is the features of the training instances, and m is the number of nearest neighbors to select.

2) Sample Generation: Set the number of samples to be generated, and then generate the specified number of samples by using the nearest-neighbor data selected as input to the imVSG model. This stage uses the imVSG model to generate virtual samples based on the neighbors identified in the previous step. A new sample z is generated by interpolating between neighbors x_n using weight α

$$z = \sum_{n=1}^{m} \alpha_n x_n. \tag{8}$$

The weights α reflect the manifold's intrinsic geometric properties and ensure that the virtual samples lie on or near the manifold learned by Isomap.

3) Sample Selection: The goal of this step is to select a representative data point from the virtual sample select representative data points. Given a minimum sample size threshold, the method is able to adaptively select for the to-be-interpreted instance the suitable data points for the instance to be interpreted, thus determining the density of its neighborhood. Once a set of virtual samples Z is generated, representative samples are selected using hierarchical clustering. This can be represented by a clustering algorithm H that partitions Z into clusters C and selects representative samples from each cluster

$$C_j = H(Z), j = 1, 2, ..., J \tag{9}$$

where J is the number of clusters formed, and representative samples are chosen by identifying the centroid or medoid of each cluster C_j.

4) Weighting Function: The weighting function π assigns weights to each of the virtual samples based on their proximity to the instance x, which can be represented as

$$W_z = \pi(z, x) = e^{(-\gamma \cdot d(x,z)^2)} \tag{10}$$

where W_z is the weight for a virtual sample z, $d(x, z)$ is the distance between x and z, and γ is a hyperparameter that controls the width of the neighborhood.

5) Feature Selection: The feature selection step aims to identify the most relevant features F that contribute to the prediction of the classifier f. This can be done using a feature selection technique such as forward selection, backward elimination, or a regularization method.

Algorithm 1: imVSG-LIME Model.

Input: Training set X_{train}, classifier f, instance to be explained x, length K, number of samples N, weighting function π

Output: Explanation model g

01: Initialize $Y = \{\cdot\}, W = \{\cdot\}, Z = \{\cdot\}, F = \{\cdot\}$
02: Neighbors $=$ SelectNeighbors(x, X_{train})
03: $Z =$ imVSG(Neighbors, N)
04: $Z =$ DataSelection(x, f, Z, t)
05: $W = W \cup \pi(z)$
06: $Y = Y \cup f(z)$
07: $F =$ FeatureSelection(Z, K)
08: return $g =$ LinearRegression(Z, Y, W, F)

6) Explanation Model Training: The final step involves training a linear regression model g using the selected virtual samples Z, their corresponding weights W, and the features F. The model is trained to approximate the classifier f within the neighborhood defined by Z

$$g(Z) = \beta_0 + \sum_{k=1}^{K} \beta_k F_k \tag{11}$$

where β_0 is the intercept, β_k is the coefficient for feature F_k, and K is the number of selected features.

The pseudo-code of the proposed imVSG-LIME model is shown in Algorithm 1.

E. Counterfactual Explanations

This section presents a novel approach that seamlessly integrates interpretability and explainability into sentiment analysis on social media platforms. The method employs a divide-and-conquer strategy to segment the search for counterfactual explanations into local feature combinations. It constructs a local greedy tree, representing interpretability, and selects optimal feature combination paths based on predetermined rules for explainability [40]. These selected paths are then interlinked to populate the feature space of the counterfactual explanation.

1) Feature Division and Local Greedy Tree Construction: The first step involves segmenting the feature space into more manageable subspaces. This can be mathematically represented as dividing the feature space F into disjoint local feature sets $F_1, F_2, ..., F_n$. For each local feature set F_i, we construct a local greedy tree T_i which evaluates the LRS for every possible feature combination C within F_i

$$\text{LRS}(C) = \sum_{j=1}^{|C|} \rho(F_{ij}, S, P) \tag{12}$$

where ρ is a scoring function that assesses the importance of a feature F_{ij} in distinguishing between the prototype sample P and the query sample S, and $|C|$ is the cardinality of the feature combination.

2) Optimal Feature Selection Path Identification: An optimal path is a sequence of features $\textbf{Path} = \{f_1, f_2, ..., f_m\}$ chosen from the local greedy tree based on a set of rules that maximize the LRS while considering the computational complexity and memory constraints. This path dictates the route taken through the feature space to construct the counterfactual explanation.

3) Counterfactual Explanation Assembly: Using the paths determined from the local greedy trees, a comprehensive counterfactual explanation is assembled by combining features from prototype sample $\mathbf{P}$ and query sample $\mathbf{Q}$. The final counterfactual explanation CE is constructed iteratively, where CE$[i]$ is determined by

$$\text{CE}[i] = \begin{cases} \mathbf{P}[i] & \text{if } \textbf{Path } [i] = 0 \\ \mathbf{Q}[i] & \text{if } \textbf{Path } [i] = 1. \end{cases} \tag{13}$$

4) Algorithm Complexity Management: As the feature space grows, the complexity of the greedy tree can increase exponentially. To manage this, a balance between local feature representation and the computational cost is maintained. This can be formalized using a complexity function $\Phi(T_i)$ that assesses the complexity of a local greedy tree T_i

$$\Phi(T_i) = \lambda \cdot |F_i| + \mu \cdot \text{nodes}(T_i). \tag{14}$$

where λ and μ are weighting parameters, $|F_i|$ is the number of features in the local set F_i, and nodes(T_i) is the number of nodes in the local greedy tree T_i. The objective is to minimize $\Phi(T_i)$ while maximizing LRS.

The generative link tree method employs several strategies to balance feature representation and computational cost when generating counterfactual explanations. It begins by segmenting the feature space into smaller subspaces, which reduces the computational complexity of searching for optimal feature combinations and allows for parallel processing. The method dynamically adjusts the depth of each local greedy tree based on the feature subspace's complexity, allowing for deeper trees in complex subspaces and shallower ones in simpler areas. To further optimize computation, pruning strategies are employed to eliminate less promising branches early in the generation process, effectively reducing the search space without significantly compromising explanation quality. The local feature representation score is computed incrementally as features are added to the combination, enabling early stopping when score improvements fall below a set threshold. The method implements a caching mechanism for intermediate results and frequently accessed feature combinations to avoid redundant computations, especially when generating multiple counterfactuals for related queries. Finally, a tunable tradeoff parameter is introduced to explicitly control the balance between feature representation quality and computational cost. This comprehensive approach allows the generative link tree method to balance thorough feature representation and manageable computational costs, facilitating the efficient generation of high-quality counterfactual explanations.

During feature combination selection, the inclusion of more features simultaneously aims to represent global features of the sample, thus enhancing interpretability. However, the increase in feature dimension may exponentially escalate the complexity of the greedy tree, posing challenges like memory overflow and prolonged explanation generation times. To address this, feature division is implemented. This process ensures local feature

Algorithm 2: Counterfactual Explanations Filling.

Input: **P**, **Q**, feature selection path sequence **PathList**
Output: Counterfactual explanations CE
Path, CE ← [ ], [ ]
// Splice feature selection path
for **P** in **PathList** do
  Path ← path.push(p)
end for
// Counterfactual explanation filling
for $i \leftarrow 1$ to Length(path) **do**
  if path$[i] == 0$
    CE.push(**P**$[i]$)
  else
    CE.push(**Q**$[i]$)
  end if
end for
return CE

representation while considering the cost of generating counterfactual interpretations. Algorithm 2 details the steps for generating counterfactual explanations from the feature selection path.

The push$(\cdot)$ operation in Algorithm 2 is used to add elements to a specific list. This operation provides control over the origin of the features that ultimately populate the counterfactual interpretation, facilitating the selection of features from both the prototype and query samples, which contributes to both interpretability and explainability.

F. Visual/Verbal Explanations

Visual explanations present the important tokens and their importance scores I_i visually using highlighting or plots, allowing users to seamlessly identify key text snippets contributing to the predicted sentiment [41].

Verbal explanations generate natural language justifications for predictions using a trained generator model

$$\exp_{\text{verbal}} = f_{\text{generator}}(\exp_{\text{unified}}) \tag{15}$$

where $f_{\text{generator}}$ is an LSTM network that takes as input the unified explanation vector $\exp_{\text{unified}}$ and generates sentences explaining the prediction in natural language.

For example, for the text "The food here is delicious!", the verbal explanation could be

> "The positive sentiment is due to the descriptive word "delicious" applauding the food quality."

Verbal explanations enhance human understandability of the model's predictions. The generator can potentially provide customized explanations based on user backgrounds using conditional training.

By supporting both visual and verbal explanations, XSA caters to different user needs and preferences for interpreting model predictions.

The XSA framework employs a multifaceted approach to transform numerical outputs into human-understandable explanations. At its core, the model assigns importance scores to input features based on their contribution to the final sentiment prediction, utilizing techniques such as LIME and SHAP. These scores are then visualized through color coding or text highlighting, creating an intuitive representation of the model's focus. To further enhance interpretability, the framework includes a natural language generation module that converts numerical scores and predictions into coherent statements, explaining the reasoning behind the sentiment analysis. The system also generates contrastive explanations, presenting "what-if" scenarios to illustrate how small input changes could alter the sentiment prediction. Numerical confidence scores are translated into qualitative statements, giving users a clear sense of the model's certainty.

The comprehensive set of explanations enabled by XSA facilities opening the black box of sentiment analysis models. Users can validate model predictions based on the presented explanations to make informed decisions regarding their appropriateness for the application context. This builds transparency and trust.

Overall, the XSA framework diagram is shown in Fig. 1.

IV. Experiment and Results Analysis

A. imVSG-LIME Model

This section first conducts experiment to analyze the efficacy of the proposed imVSG-LIME model of XSA framework in enabling explainability specifically for sentiment analysis focused on diverse social media data.

1) Dataset
2) For this experiment, we employ a carefully curated dataset of tweets systematically sampled from the Twitter platform [2]. The dataset is for aspect-based sentiment analysis, which delves into the nuanced sentiments expressed in the text concerning specific aspects or categories. This dataset comprises approximately 15 000 tweets, each meticulously annotated for sentiment polarities. The tweets in this dataset are categorized based on their sentiment towards various aspects commonly discussed in restaurant reviews.

Each tweet in the dataset is carefully labeled with one of three possible sentiment polarities: positive, negative, or neutral. These labels provide a detailed and granular understanding of how users on Twitter perceive and express their opinions regarding different aspects of restaurant experiences.

To ensure robust model evaluation, we divide the dataset into three distinct subsets: training, validation, and test sets. The total is 15 000 tweets, 11 000 tweets for the training set, and is used for training the imVSG-LIME model, providing the model with diverse examples to learn from and adapt to different nuances in Twitter sentiment data. The validation set includes 2000 tweets and is used to fine-tune model hyperparameters and assess their performance during training. It helps in preventing overfitting and optimizing model generalization. The test set includes 2000 tweets for evaluating the model's performance after training, providing a realistic evaluation of how well the model can generalize to unseen data and make sentiment predictions based on aspect-focused analysis.

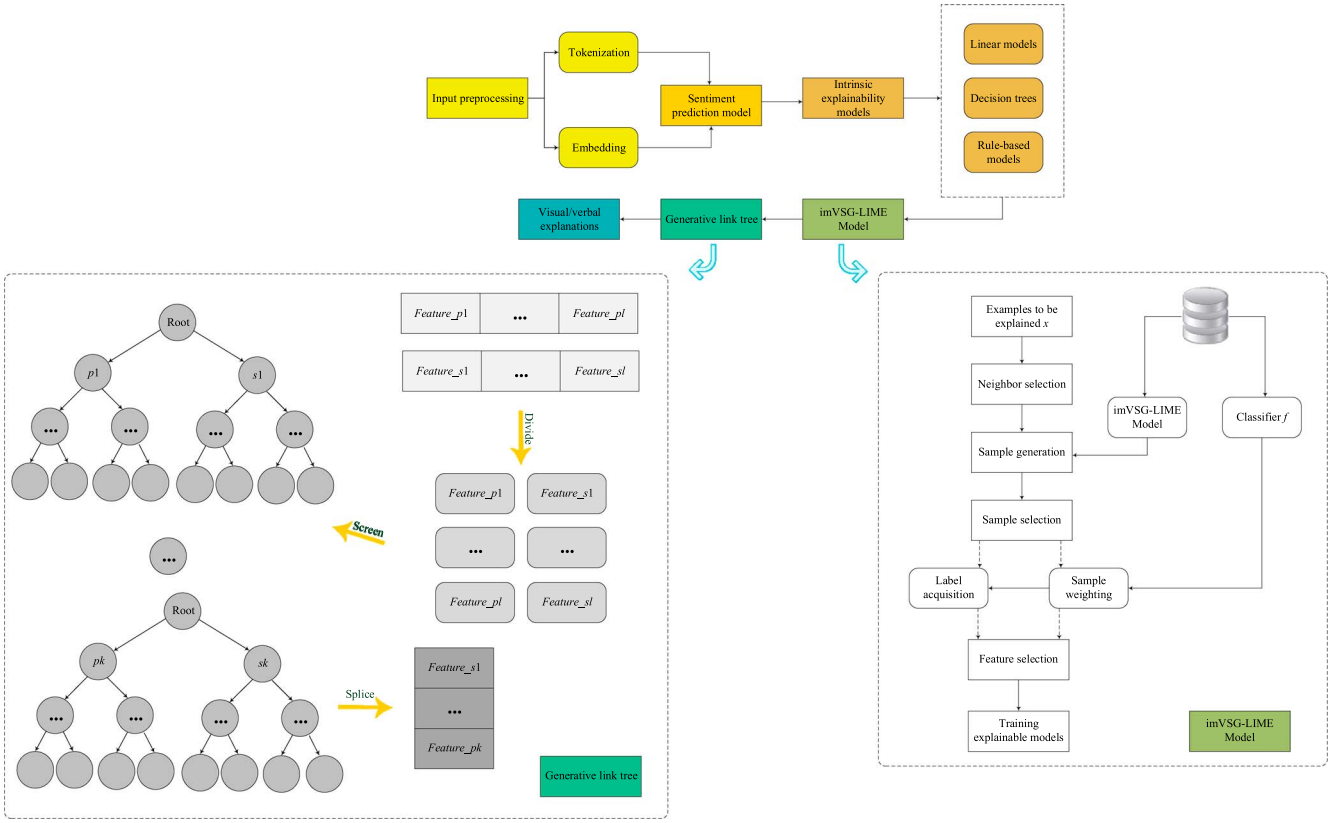

Fig. 1. XSA framework.

The key mathematical parameters used in the experiments with values are listed in Table I below:

The parameters γ, λ, and μ were set by tuning them in a grid search over values {0.1, 0.3, 0.5, 0.7, 0.9} using fivefold cross-validation. The values resulting in optimal performance on the validation set were selected.

1) *Evaluation Metrics:*
   a) Aspect sentiment F1 score measures accuracy of predicted sentiment for each aspect based on the F1 score for positive and negative classes. The accuracy measures how often the model's sentiment predictions match the true sentiment labels. F1 score calculates the harmonic mean of precision and recall for the positive and negative sentiment classes.
   b) *Explanation Plausibility:* It evaluates how credible the explanations are for social media sentiment predictions based on user surveys on a 1–5 scale.
   c) *Explanation Faithfulness:* It quantifies agreement between explanations and model predictions for social media sentiment using Pearson correlation between explanation importance scores and prediction probabilities.
   d) *User Trust Score:* It captures how much users trust the model's sentiment predictions on social posts based on its explanations, from 1 to 5 based on user surveys.

We use BERT [35], LIME-HPO [30], KGDL-SA [31], VADER-LIME [32], SHAP-XGBoost [33], AEC-LSTM [34], BiERU [24], ABCDM [25], and the proposed imVSG-LIME model of XSA framework for comparison. Table II shows the aspect sentiment analysis results on the Twitter dataset.

TABLE I
MATHEMATICAL PARAMETER SETTINGS

Parameter	Description	Value
γ	Neighborhood width	0.5
λ	Weights sample complexity	0.3
μ	Weights tree complexity	0.7
K	Number of features for explanation model	10
N	Number of virtual samples generated	1000
m	Number of nearest neighbors for imVSG	100

Next, we use a dataset of YouTube comments labeled for sentiment analysis, which comprises 240 000 comments labeled with sentiment polarity (positive, negative). Specifically, out of the 240 000 comments, 192 000 comments are allocated for training the models, 24 000 comments are used for validation during the model development process, and another 24 000 comments are reserved for final testing and performance evaluation. This split enables them to train the models using extensive data while maintaining separate subsets for model selection and evaluation. The comments discuss diverse topics, including sports, entertainment, politics, products, and movies. Table III

TABLE II
ASPECT SENTIMENT ANALYSIS RESULTS ON TWITTER DATASET

Model	Service F1	Food F1	Staff F1	Price F1	Ambience F1	Waiting F1	Explanation Plausibility	Explanation Faithfulness	User Trust Score
BERT	0.784	0.835	0.736	0.712	0.767	0.773	-	-	3.126
LIME-HPO	0.768	0.819	0.713	0.693	0.748	0.757	3.178	0.671	3.315
KGDL-SA	0.786	0.824	0.765	0.704	0.759	0.749	3.554	0.628	3.847
VADER-LIME	0.753	0.803	0.719	0.728	0.766	0.715	3.668	0.683	3.908
SHAP-XGBoost	0.775	0.821	0.722	0.741	0.752	0.76	3.526	0.722	3.744
AEC-LSTM	0.809	0.857	0.767	0.799	0.803	0.779	4.012	0.825	3.815
BiERU	0.812	0.884	0.778	0.846	0.811	0.789	4.216	0.799	3.946
ABCDM	0.836	0.903	0.826	0.839	0.824	0.866	4.268	0.857	4.257
imVSG-LIME (XSA)	**0.853**	**0.922**	**0.918**	**0.857**	**0.879**	**0.891**	**4.334**	**0.916**	**4.841**

TABLE III
DOCUMENT SENTIMENT ANALYSIS RESULTS ON YOUTUBE COMMENTS

Model	Accuracy	F1	Explanation Plausibility	Explanation Faithfulness	User Trust Score
BERT	0.825	0.832	-	-	3.276
LIME-HPO	0.836	0.863	3.457	0.785	3.648
KGDL-SA	0.877	0.862	3.205	0.716	3.479
VADER-LIME	0.879	0.864	3.403	0.791	3.603
SHAP-XGBoost	0.903	0.882	3.598	0.805	3.267
AEC-LSTM	0.916	0.898	4.012	0.864	4.295
BiERU	0.921	0.897	4.215	0.917	4.365
ABCDM	0.905	0.924	4.087	0.924	4.624
imVSG-LIME (XSA)	**0.928**	**0.937**	**4.561**	**0.932**	**4.789**

shows the document sentiment analysis results on the YouTube comments dataset.

As can be seen from Tables II and III, the proposed imVSG-LIME model likely produces higher quality explanations, as indicated by the higher explanation plausibility scores in the results, meaning that the model's reasoning is more understandable to users, which is essential for trust and transparency. With high scores in explanation plausibility and user trust, the proposed imVSG-LIME model fosters greater trust among users. This is critical for applications where understanding model predictions affects decision-making processes. While the proposed imVSG-LIME model's superior F1 scores across various aspects of sentiment analysis suggest that it provides explanations while maintaining high prediction accuracy.

B. Generative Link Tree

To verify the data fidelity of counterfactual explanations generated by the generative link tree method, we propose data fidelity metric. Counterfactual explanations should have high fidelity with respect to the original data, and should comply with validity and pan-adaptability in experiments. Validity means that the categories of counterfactual explanations should be of the desired categories, and the experiments should rely on the validation model, i.e., the model on which the generated counterfactual explanations are based, and its categorization results of the counterfactual explanations should be of the desired categories. A classification model that is different from the validation model used to generate the counterfactual explanations is called a "third-party model," and its classification of counterfactual explanations should have a high classification accuracy, F1-score, and other metrics. Since we need to rely on the validation model for validation in the generation process, and screen out the counterfactual explanations classified by the validation model into the desired categories, the counterfactual explanations can usually satisfy the validity under the condition of ensuring the validation model remains unchanged. For the pan-adaptability, this article proposes the evaluation metric of data fidelity, which is defined as follows:

$$\mathrm{DF} = \frac{\sum_{i=1}^{n} w_i \times p_i}{\sum_{i=1}^{n} w_i} \tag{16}$$

where w_i is the corresponding weight of the third-party model, expressed as its accuracy in classifying the original data, measured as F1-score. p_i denotes the degree to which the third-party model recognizes the truthfulness of the counterfactual explanations of the data, expressed as the degree to which the third-party model accurately classifies the counterfactual explanations, measured as F1-score.

Intuitively, DF evaluates how accurately the third-party models can classify the generated counterfactual explanations. The weight w_i controls the relative importance assigned to each third-party model based on its competence in classifying real samples. A higher DF implies counterfactual explanations closely reflect true data distribution since even external third-party models not involved in their generation can categorize them accurately. This demonstrates the broader credibility of the counterfactual explanations.

This section uses Twitter, YouTube comments, Yelp, and Amazon comments datasets to generate counterfactual explanations by randomly selecting query samples, repeating the same experiments in ten groups to take the mean, target coding the categorical features in the samples, and generating counterfactual explanations by using the generative link tree and baseline methods, respectively. Using the data fidelity proposed in this article as the evaluation metric, several sets of experiments are conducted from the perspectives of generating multiple counterfactual explanations for single query samples and generating multiple counterfactual explanations for multiple query samples, respectively. In these experiments, the features in the YouTube comments dataset are divided into two groups of four, and the features in the Twitter dataset are divided into four groups of five. Additionally, to explore the effect of different granularity of segmentation on the data fidelity, we also conduct experiments with different segmentation on Twitter dataset which has a larger number of features.

The XSA framework employs a versatile approach to handle the diverse characteristics of different social media platforms. For each platform, custom preprocessing steps are implemented to address unique features, such as handling hashtags and mentions on Twitter, processing video metadata on YouTube, or dealing with structured review data on Yelp and Amazon. The framework dynamically adjusts its vocabulary and embedding space to account for platform-specific jargon and expressions through transfer learning techniques. Length normalization techniques are incorporated into the feature extraction process to address varying text lengths across platforms. For platforms with rich multimodal content such as YouTube, additional modules are integrated to process and combine information from images and videos alongside text. The framework also considers temporal aspects, accounting for the timing and sequence of comments or the evolution of conversations over time. Platform-specific user interaction patterns, such as likes, shares, and retweets, are incorporated as additional features to provide context for sentiment analysis.

This article chooses the random forest model as the verification model for counterfactual explanations to verify whether the generated counterfactual explanations are the target category. Additionally, commonly used machine learning models, such as decision trees (DT), naive Bayes (NB), and others, are selected as third-party models to evaluate the data authenticity of counterfactual explanations. The weight of the third-party model is determined using tenfold cross-validation.

Suppose the classification result of the third-party model on the counterfactual explanation is regarded as its recognition of the authenticity of the counterfactual explanation data. In that case, the weight can be understood as the authority of the corresponding "judge." After tenfold cross-validation, this study determines the weights of five third-party models, and the results are shown in Table IV.

In this article, the Dice framework [42] is chosen as the baseline approach, and comparative experiments are conducted from the perspective of generating multiple counterfactual explanations for a single query sample and multiple counterfactual explanations for multiple query samples. The Dice framework is a methodology used in XAI to generate counterfactual explanations. In this article, five to ten counterfactual explanations are generated for a single query sample, and a third-party model is utilized to verify the authenticity of their data. The experimental results are shown in Fig. 2(a)–2(e), which give the F1-scores of the counterfactual explanations categorized by the different models.

TABLE IV
VALIDATION WEIGHTS OF THE THIRD-PARTY MODEL

Model	YouTube Comments Dataset	Twitter Dataset	Yelp Dataset	Amazon Comments Dataset
KNN	0.74	0.67	0.71	0.74
MLP	0.76	0.68	0.73	0.75
SVM	0.75	0.70	0.74	0.73
DT	0.7	0.65	0.69	0.68
NB	0.74	0.71	0.74	0.72

In Fig. 2, Y denotes the YouTube comments dataset, and the number denotes the number of counterfactual interpretations generated, e.g., "Y5" denotes that a query sample in the YouTube comments dataset generates five counterfactual explanations. Dice-r, Dice-g, and Dice-k denote the three machine learning based methods provided in the Dice framework. According to (18), the data fidelity of counterfactual explanations generated by different methods is obtained, as listed in Table V, and the bolds indicate the highest data fidelity under the corresponding tasks.

As observed in Fig. 2, the proposed GLT method in the XSA framework achieves significantly higher F1 scores across multiple third-party models. The consistent superiority in F1 scores across third-party models exhibits the ability of the GLT method to produce counterfactual explanations that closely reflect the actual data distribution, enabling even models not involved in the counterfactual generation process to accurately classify the explanations, demonstrating pan-adaptability. In contrast, the lower F1 scores achieved by DICE method variants indicate that the counterfactuals deviate more from the actual data, leading to poorer sentiment identification by external third-party models. Overall, the results validate the capability of the proposed approach within the XSA framework to generate counterfactual explanations that reliably capture authentic sentiments, bolstering trust in the system. The high pan-adaptability opens up broader reliable utilization of the counterfactuals beyond the original verification model.

In addition, this study conducts experiments from the perspective of generating counterfactual explanations for multiple query samples. Five query samples were randomly selected ten times, ten counterfactual explanations were generated for each sample, and the data authenticity of these counterfactual explanations was verified. The F1-score of the third-party model classifying counterfactual explanations is shown in Fig. 3(a) and 3(b). By analyzing the F1 scores, we can gain insights into the effectiveness of the counterfactual explanations and the reliability of the employed third-party classification model. This evaluation helps assess the overall quality of the counterfactual generation process and the extent to which the generated explanations align

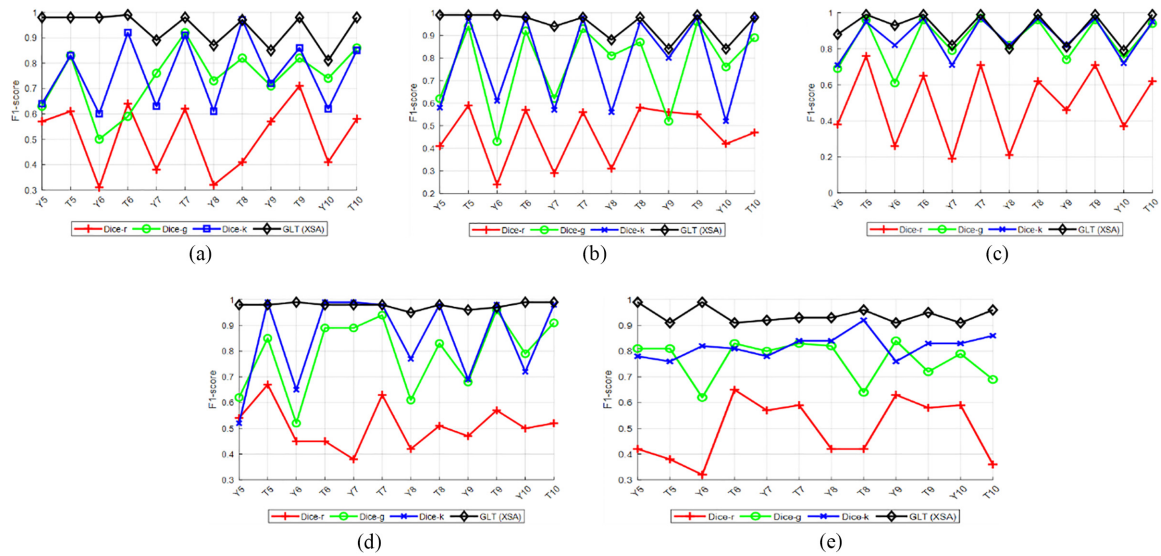

Fig. 2. Classification results of third-party models. (a) KNN. (b) MLP. (c) SVM. (d) DT. (e) NB.

TABLE V
DATA FIDELITY OF MULTIPLE COUNTERFACTUAL EXPLANATIONS

Methods	Y5	T5	Y6	T6	Y7	T7	Y8	T8	Y9	T9	Y10	T10
Dice-r	0.471	0.620	0.343	0.631	0.384	0.621	0.363	0.546	0.552	0.623	0.476	0.523
Dice-g	0.712	0.911	0.565	0.916	0.766	0.955	0.784	0.729	0.726	0.905	0.765	0.884
Dice-k	0.653	0.922	0.723	0.952	0.692	0.964	0.717	0.970	0.776	0.954	0.678	0.953
Generative link tree (XSA)	0.984	0.985	0.985	0.986	0.932	0.986	0.986	0.986	0.896	0.992	0.891	0.991

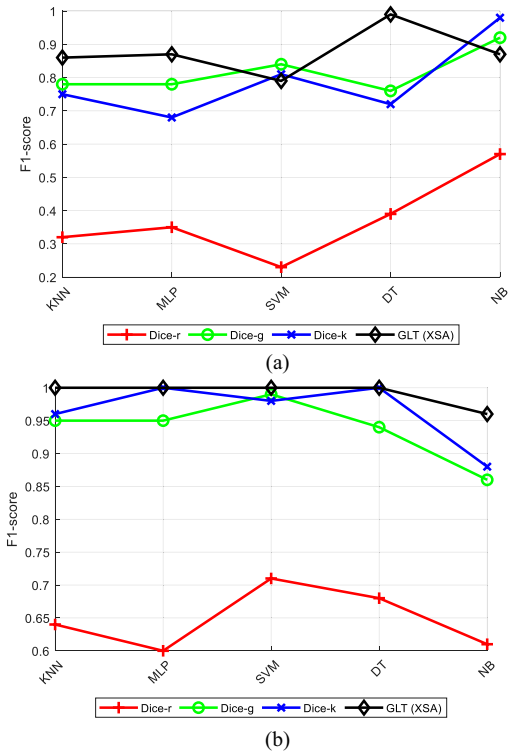

Fig. 3. Classification results. (a) YouTube comments dataset. (b) Twitter dataset.

with the underlying decision-making mechanisms of the machine learning model being explained.

The high data fidelity of the proposed generative link tree method within the XSA framework suggests that this model is capable of accurately capturing and reflecting the true sentiment as expressed in the data. High data fidelity in the results of the generative link tree method can lead to increased trust from users and stakeholders. When users can see that the model's predictions closely match the real-world data, they are more likely to trust and rely on the system's outputs. A model that accurately reflects data realities makes its inner workings more interpretable. When the model's predictions are highly faithful to the data, it is easier to understand why the model makes certain decisions, thereby improving explainability. Decision-makers can use the insights derived from a high data fidelity model with confidence, knowing that the sentiments analyzed reflect actual opinions and feelings expressed in the source material. For applications that interact directly with users, such as customer feedback analysis, high data fidelity ensures that the sentiment analysis aligns with users' intended meanings, leading to a better overall user experience. High data fidelity means that any discrepancies between the model's predictions and actual sentiments are likely due to the model itself rather than the data. This clarity can help developers focus their efforts on refining the model for even better performance. In XAI, an accurate model such as generative link tree reduces the risk of misrepresenting

TABLE VI
DATA FIDELITY OF MULTIPLE COUNTERFACTUAL EXPLANATIONS

Datasets	Dice-r	Dice-g	Dice-k	Generative Link Tree (XSA)
YouTube comments	0.372	0.826	0.756	0.889
Twitter	0.653	0.942	0.971	0.994
Yelp	0.468	0.886	0.876	0.912
Amazon	0.532	0.907	0.953	0.965

sentiments, which can be crucial for ethical AI practices, ensuring that all voices are heard and accurately represented. In summary, the high data fidelity of the generative link tree method within the XSA framework contributes significantly to the effectiveness and reliability of sentiment analysis applications, making them more useful for both end-users and model developers in the pursuit of clear, fair, and actionable sentiment insights.

The results of data fidelity after weighted are listed in Table VI.

V. CONCLUSION

This article focuses on improving explainability in sentiment analysis models for social media platforms using XAI. We propose the XSA framework incorporating intrinsic and posthoc XAI methods to generate explanations for model predictions. Specifically, to address the lack of local fidelity and instability in interpretations caused by LIMEs random perturbation sampling, we introduce a new model-agnostic interpretation approach, utilizing imVSG based on manifold learning instead of LIME's random sampling to generate more reliable synthetic samples for local explanation. Additionally, we present a generative link tree method to create high-fidelity counterfactual explanations by leveraging training data examples and a divide-and-conquer greedy strategy. Experiments on benchmark social media datasets demonstrate XSAs ability to provide local token and aspect-level explanations while maintaining competitive sentiment analysis performance. Analyses reveal enhanced model interpretability and improved user trust in the system. Overall, the results highlight the potential of XAI techniques to open the black box of modern deep learning sentiment models for social media and improve their adoption by building trust. The proposed XSA framework is valuable for developing accurate, transparent, and trustworthy sentiment analysis systems.

However, some limitations of this study present opportunities for future work. First, we focus only on benchmark social media datasets from limited platforms. Further evaluation of diverse social media sources can help generalize XSAs benefits. Second, we consider only English language sentiment analysis. Expanding to other languages such as Chinese and Hindi, is an important future direction. Finally, studying the effects of diverse explanation styles and visualizations on user trust and mental models can further optimize XSAs interface for human consumption. To conclude, explainability is essential for trustworthy sentiment analysis on social media. As this work demonstrated, leveraging XAI techniques can help open the black box of modern deep learning models to build transparent and fair sentiment analysis systems. The proposed XSA framework offers a valuable step in this direction. Further research can focus on generalizing XSA across languages and diverse social media platforms, optimizing explanations for human mental models, and combining insights from multiple XAI methods.

REFERENCES

[1] R. Das and T. D. Singh, "Multimodal sentiment analysis: A survey of methods, trends, and challenges," *ACM Comput. Surv.*, vol. 55, no. 135, Oct. 2023, Art. no. 270, doi: 10.1145/3586075.

[2] H. Rehioui and A. Idrissi, "New clustering algorithms for twitter sentiment analysis," *IEEE Syst. J.*, vol. 14, no. 1, pp. 530–537, Mar. 2020, doi: 10.1109/JSYST.2019.2912759.

[3] M. Ben Hajhmida and O. Oueslati, "Predicting mobile application breakout using sentiment analysis of Facebook posts," *J. Inf. Sci.*, vol. 47, no. 4, pp. 502–516, Aug. 2021, doi: 10.1177/0165551520917099.

[4] D. Ma, Y. Wang, J. Ma, and Q. Jin, "SGNR: A social graph neural network based interactive recommendation scheme for E-commerce," *Tsinghua Sci. Technol.*, vol. 28, no. 4, pp. 786–798, Aug. 2023, doi: 10.26599/TST.2022.9010050.

[5] Y. Bie, Y. Yang, and Y. L. Zhang, "Fusing syntactic structure information and lexical semantic information for end-to-end aspect-based sentiment analysis," *Tsinghua Sci. Technol.*, vol. 28, no. 2, pp. 230–243, Apr. 2023, doi: 10.26599/TST.2021.9010095.

[6] R. Zeng et al., "CNN-based broad learning for cross-domain emotion classification," *Tsinghua Sci. Technol.*, vol. 28, no. 2, pp. 360–369, Apr. 2023, doi: 10.26599/TST.2022.9010007.

[7] J. J. Li, Y. Wang, and C. L. Liu, "Spatial effect of market sentiment on housing price: Evidence from social media data in China," *Int. J. Strateg. Prop. Manag.*, vol. 26, no. 1, pp. 72–85, Feb. 2022, doi: 10.3846/ijspm.2022.16255.

[8] M. M. Aguero-Torales, J. I. A. Salas, and A. G. Lopez-Herrera, "Deep learning and multilingual sentiment analysis on social media data: An overview," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107373, doi: 10.1016/j.asoc.2021.107373.

[9] P. F. Yu, W. A. Tan, W. A. Niu, and B. Shi, "Aspect-location attention networks for aspect-category sentiment analysis in social media," *J. Intell. Inf. Syst.*, vol. 61, no. 2, pp. 395–419, Feb. 2023, doi: 10.1007/s10844-022-00760-2.

[10] H. Lee, N. Lee, H. Seo, and M. Song, "Developing a supervised learning-based social media business sentiment index," *J. Supercomput.*, vol. 86, no. 5, pp. 395–419, May 2020, doi: 10.1007/s11227-018-02737-x.

[11] L. He, T. J. Yin, and K. Zheng, "They may not work! An evaluation of eleven sentiment analysis tools on seven social media datasets," *J. Biomed. Inform.*, vol. 132, Aug. 2022, Art. no. 104142, doi: 10.1016/j.jbi.2022.104142.

[12] Z. G. Jin, M. Y. Tao, X. F. Zhao, and Y. Hu, "Social media sentiment analysis based on dependency graph and co-occurrence graph," *Cognit. Comput.*, vol. 14, no. 3, pp. 1039–1054, May 2022, doi: 10.1007/s12559-022-10004-8.

[13] J. Yuan, F. R. Lin, and H. Y. Kim, "Exploring artistic embeddings in service design: A keyword-driven approach for artwork search and recommendations," *Tsinghua Sci. Technol.*, vol. 29, no. 5, pp. 1580–1592, Oct. 2024, doi: 10.26599/TST.2023.9010118.

[14] T. L. Liu, J. W. Wan, X. B. Dai, F. Liu, Q. Z. You, and J. B. Luo, "Sentiment recognition for short annotated GIFs using visual-textual fusion," *IEEE Trans. Multimed.*, vol. 22, no. 4, pp. 1098–1110, Apr. 2020, doi: 10.1109/TMM.2019.2936805.

[15] R. Wadawadagi and V. Pagi, "Sentiment analysis with deep neural networks: Comparative study and performance assessment," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 6155–6195, Dec. 2020, doi: 10.1007/s10462-020-09845-2.

[16] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[17] M. B. Shelke, D. D. Sawant, C. B. Kadam, K. Ambhure, and S. N. Deshmukh, "Marathi SentiWordNet: A lexical resource for sentiment analysis of Marathi," *Concurr. Comput. Pract. Exp*,. vol. 35, no. 2, Jan. 2023, doi: 10.1002/cpe.7497.

[18] A. Nazir, Y. Rao, L. W. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *Knowl.-Based*

Syst., vol. 13, no. 3, pp. 845–863, Jun. 2022, doi: 10.1109/TAFFC.2020.2970399.

[19] M. Xu, F. F. Liang, X. Y. Su, and C. Fang, "CMJRT: Cross-modal joint representation transformer for multimodal sentiment analysis," *IEEE Access*, vol. 10, pp. 131671–131679, Jan. 2023, doi: 10.1109/ACCESS.2022.3219200.

[20] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manage*, vol. 58, no. 1, Jan. 2021, Art. no. 102435, doi: 10.1016/j.ipm.2020.102435.

[21] M. S. Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, "All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework," *IEEE Trans. Affect.*, vol. 13, no. 1, pp. 285–297, Jan. 2022, doi: 10.1109/TAFFC.2019.2926724.

[22] K. Zhang, Q. Liu, H. Qian, B. Xiang, Q. Cui, and E. H. Chen, "EATN: An efficient adaptive transfer network for aspect-level sentiment analysis," *IEEE Trans Knowl. Data Eng.*, vol. 35, no. 1, pp. 377–389, Jan. 2023, doi: 10.1109/TKDE.2021.3075238.

[23] B. Liang, H. Su, L. Gui, E. Cambria, and R. F. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowl.-Based Syst.*, vol. 235, pp. 107643, Jan. 2022, doi: 10.1016/j.knosys.2021.107643.

[24] W. Li, W. Shao, S. X. Ji, and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, pp. 73–82, Jan. 2022, doi: 10.1016/j.neucom.2021.09.057.

[25] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Gener. Comput. Syst.*, vol. 115, pp. 279–294, Feb. 2021, doi: 10.1016/j.future.2020.08.005.

[26] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, Apr. 2020, doi: 10.1109/ACCESS.2020.2969854.

[27] M. Zolanvari, Z. B. Yang, K. Khan, R. Jain, and N. Meskin, "TRUST XAI: Model-agnostic explanations for AI with a case study on IIoT security," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2967–2978, Feb. 2023, doi: 10.1109/JIOT.2021.3122019.

[28] G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu, "On the tractability of SHAP explanations," *J. Artif. Intell. Res.*, vol. 74, pp. 851–886, Jul. 2022, doi: 10.1613/jair.1.13283.

[29] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI," *Inf. Fusion*, vol. 71, pp. 28–37, Jul. 2021, doi: 10.1016/j.inffus.2021.01.008.

[30] J. Jiarpakdee, C. Tantithamthavorn, H. K. Dam, and J. Grundy, "An empirical study of model-agnostic techniques for defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 166–185, Jan. 2022, doi: 10.1109/TSE.2020.2982385.

[31] F. A. Lovera, Y. C. Cardinale, and M. N. Homsi, "Sentiment analysis in Twitter based on knowledge graph and deep learning classification," *Electronics*, vol. 10, no. 22, Nov. 2021, Art. no. 2739, doi: 10.3390/electronics10222739.

[32] R. Jain et al., "Explaining sentiment analysis results on social media texts through visualization," *Multim. Tools Appl.*, vol. 82, no. 15, pp. 22613–22629, Feb. 2023, doi: 10.1007/s11042-023-14432-y.

[33] Z. Q. Li, "Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost," *Comput. Environ. Urban Syst.*, vol. 96, Sep. 2022, Art. no. 101845, doi: 10.1016/j.compenvurbsys.2022.101845.

[34] F. L. Huang, X. L. Li, C. A. Yuan, S. C. Zhang, J. L. Zhang, and S. J. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4432–4345, Sep. 2022, doi: 10.1109/TNNLS.2021.3056664.

[35] A. Pimpalkar and R. J. R. Raj, "MBiLSTMGloVe: Embedding GloVe knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis," *Exp. Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117581, doi: 10.1016/j.eswa.2022.117581.

[36] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[37] K. Hamm, N. Henscheid, and S. J. Kang, "Wassmap: Wasserstein isometric mapping for image manifold learning," *SIAM J. Math. Data Sci.*, vol. 5, no. 2, pp. 475–501, Oct. 2023, doi: 10.1137/22M1490053.

[38] J. Lai, X. D. Wang, Q. Xiang, Y. F. Song, and W. Quan, "Multilayer discriminative extreme learning machine for classification," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 6, pp. 2111–2125, Jun. 2023, doi: 10.1007/s13042-022-01749-7.

[39] J. Dieber and S. Kirrane, "A novel model usability evaluation framework (MUsE) for explainable artificial intelligence," *Inf. Fusion*, vol. 81, pp. 143–153, May 2022, doi: 10.1016/j.inffus.2021.11.017.

[40] D. D. Niu, B. Liu, M. H. Yin, and Y. P. Zhou, "A new local search algorithm with greedy crossover restart for the dominating tree problem," *Expert Syst. Appl.*, vol. 229, 2023, Art. no. 120353, doi: 10.1016/j.eswa.2023.120353.

[41] Z. Chen, F. Silvestri, G. Tolomei, J. Wang, H. Zhu, and H. Ahn, "Explain the explainer: Interpreting model-agnostic counterfactual explanations of a deep reinforcement learning agent," *IEEE Trans. Artif. Intell.*, vol. 99, pp. 1–15, 2022, doi: 10.1109/TAI.2022.3223892.

[42] G. Casale and C. Li, "Enhancing big data application design with the DICE framework," in *Proc. ESOCC*, 2017, pp. 164–168.