



ARTICLE

## Unleashing User Requirements from Social Media Networks by Harnessing the Deep Sentiment Analytics

Deema Mohammed Alsekait<sup>1,\*</sup>, Asif Nawaz<sup>2</sup>, Ayman Nabil<sup>3</sup>, Mehwish Bukhari<sup>2</sup> and Diaan Salama Abdelminaam<sup>3,4,5,6,\*</sup>

<sup>1</sup>Department of Computer Science and Information Technology, Applied College, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

<sup>2</sup>Computer Sciences and Information Technology, University of Institute Information Technology, PMAS-Arid Agriculture University, Rawalpindi, P.O Box 46000, 46300, Pakistan

<sup>3</sup>Computer Science Department, Faculty of Computers Science, Misr International University, Cairo, 11800, Egypt

<sup>4</sup>MEU Research Unit, Middle East University, Amman, 11831, Jordan

<sup>5</sup>Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha, 13511, Egypt

<sup>6</sup>Jadara Research Center, Jadara University, Irbid, 21110, Jordan

\*Corresponding Authors: Deema Mohammed Alsekait. Email: dmalsekait@pnu.edu.sa; Diaan Salama Abdelminaam. Email: diaa.salama@miuegypt.edu.eg

Received: 17 March 2024 Accepted: 07 June 2024 Published: 17 July 2024

### ABSTRACT

The article describes a novel method for sentiment analysis and requirement elicitation from social media feedback, leveraging advanced machine learning techniques. This innovative approach automates the extraction and classification of user requirements by analyzing sentiment in data gathered from social media platforms such as Twitter and Facebook. Utilizing APIs (Application Programming Interface) for data collection and Graph-based Neural Networks (GNN) for feature extraction, the proposed model efficiently processes and analyzes large volumes of unstructured user-generated content. The preprocessing pipeline includes data cleaning, normalization, and tokenization, ensuring high-quality input for the sentiment analysis model. By classifying user feedback into requirement and non-requirement categories, the model achieves significant improvements in capturing user sentiments and requirements. The experimental results demonstrate that the model outperforms existing benchmark methods with an accuracy of 95.02%, precision of 94.74%, and recall of 94.53%. These metrics underscore the model's effectiveness in identifying and classifying user requirements accurately. The authors illustrate the proposed methodology through extensive validation and impact analysis, highlighting its effectiveness in dynamically adapting to evolving user feedback. This approach enhances the agility and user-centered nature of software development, ensuring more responsive and accurate requirement elicitation. The novelty of this research lies in the integration of automated feedback collection, advanced preprocessing techniques, and the use of GNN for feature extraction. This combination allows the model to consider the complex and interconnected data structures typical of social media content, leading to superior performance in sentiment analysis and requirement elicitation. The new method's effectiveness is further confirmed by calculations of performance metrics such as precision, recall, and F1-score. The proposed methodology's practical implications are vast, offering a potent solution for agile and user-centered software development practices. By dynamically integrating real-time user feedback, the model supports continuous software improvement and adaptation, making it highly relevant for rapidly changing user preferences and requirements. This research contributes significantly to the fields of sentiment analysis, machine



learning, and software engineering, providing a robust framework for future studies and practical applications in requirement elicitation from social media.

#### KEYWORDS

Requirements elicitation; Twitter API; Facebook; deep learning; text classification; social media networks

---

## 1 Introduction

One of the most essential steps in requirements engineering is requirements elicitation, which tries to collect user needs and preferences. To get user input, it uses a variety of methodologies, including interviews, surveys, and Joint Application Development (JAD) sessions. During maintenance, online user input gleaned from sites like the App Store and Play Store reviews is critical to software progress. Positive user input may point to potential new software requirements, while negative feedback may propose enhancing already-existing functionality. Positive comments, for instance, recommend improving the software's usability in some areas. Negative comments could draw attention to problems users have with the program [1].

Although online user reviews from app stores offer insightful information, they are sometimes insufficient because of the ubiquity of user reviews on social networking sites. Millions of people in Thailand use social media sites like Facebook, Instagram, and Twitter alone. Maleknaz et al.'s previous research on needs elicitation from the App Store, Play Store, and Twitter revealed that 12.89% of requirements were elicited via Twitter, demonstrating the value of customer input on social media. Facebook, which has four times as many users as Twitter, is still an untapped resource for valuable user input. Users commonly express their insights and opinions on software products through Facebook fan pages [2].

There exists a need for different techniques to get needed information from Facebook and Twitter. While site scraping is required for Facebook since Facebook APIs are not publically accessible, Tweepy APIs are utilized to gather user input from Twitter. Effective software evolution can be hampered by insufficient ways to gather needs from specific social media sites. This study suggests a method to overcome this problem: using the Naive Bayes machine learning algorithm to extract requirements from user input on Twitter and Facebook. This study report also offers an exhaustive explanation of the instruments used. Two primary research issues are addressed in this study:

**Research Question 1:** How do density requirements appear on Facebook and Twitter?

**Research Question 2:** What degree of accuracy does the suggested method have?

This research makes several significant contributions to the fields of requirements engineering and software development by leveraging social media analytics:

**Development of a Novel Sentiment Analysis Model:** We introduce an innovative deep learning-based model for sentiment analysis that effectively harnesses user feedback from social media platforms like Twitter and Facebook. This model goes beyond traditional requirement elicitation methods by automating the collection and analysis of user-generated content.

**Enhanced Data Preprocessing Techniques:** Recognizing the challenges associated with the unstructured nature of social media data, this study develops advanced data preprocessing methods. These include a robust cleaning process to remove noise and irrelevant information, ensuring high-quality data input into the sentiment analysis model.

**Utilization of Graph-based Neural Networks (GNN):** Our research employs GNNs for feature extraction from social media texts, which is pivotal for accurately capturing the underlying sentiments of users. This approach allows the model to consider the complex and interconnected data structures typical of social media content.

**Demonstration of Superior Model Performance:** The proposed model achieves an accuracy of 95.02%, significantly outperforming benchmark methods in sentiment analysis for requirement elicitation. This high accuracy demonstrates the model's effectiveness in identifying and classifying user requirements from social media feedback.

**Practical Implications for Agile and User-Centered Software Development:** By providing a tool that can dynamically extract and categorize user requirements from ongoing social media discussions, our model supports more agile and user-centered approaches to software development. This contribution is precious in environments where user preferences and requirements evolve rapidly.

**Extensive Validation and Impact Analysis:** We conducted a thorough analysis of the model's performance across various metrics, including Precision, Recall, and F-score. The results underscore the robustness of the approach and its potential to impact software development practices by integrating real-time user feedback into the development process.

The rest of the paper is divided into the following sections: [Section 2](#) discusses relevant work, the proposed method is described in [Section 3](#), Experimental results and analysis are given in [Section 4](#), and [Section 5](#) provides a conclusion and future work.

## 2 Literature Review

This section provides a comprehensive discussion of the existing literature. To solve communication and teamwork challenges in distributed agile software development utilizing blockchain-based systems, the study "Privacy Requirements Elicitation: A Systematic Literature Review and Perception Analysis of IT Practitioners" focuses on the literature and the perceptions of IT practitioners. Agile Plus, a blockchain-based system that executes smart contracts on a personal Ethereum blockchain, is the solution suggested by Canedo et al. to enhance Design Thinking's transparency, collaboration, communication, traceability, security, and trust. To solve the blockchain's scalability problem, the system leverages the Interplanetary File System (IPFS) as off-chain storage. Customers and developers are promptly penalized by intelligent contracts for lost or late payments and improper performance of obligations. According to experimental findings, Agile Plus enhanced trust concerns in Design Thinking and increased traceability, communication, cooperation, and security [3].

The main objective of this paper [4] is to address the challenges in the requirements elicitation process in software engineering, particularly focusing on the difficulty of ensuring that elicited requirements possess desired attributes such as completeness, correctness, and consistency. To achieve this, the paper proposes and implements a measurement program based on the ISO/IEC/IEEE 15939:2017 standard. This program includes a set of eight measures aimed at quantifying and improving the quality of preliminary requirements. A case study in a small-sized software organization demonstrates that the proposed measurement program positively influences the quality of the elicited preliminary requirements.

The paper “Guidelines Adopted by agile teams in privacy requirements elicitation after the Brazilian General Data Protection Law (LGPD) Implementation” by Canedo et al. aims to investigate the impact of the LGPD on software development teams two years after its implementation. The authors conducted a systematic literature review, a survey with 53 IT practitioners, and semi-structured interviews with ten practitioners to examine the actions taken by organizations regarding the LGPD in software development. The study found that most agile teams and Brazilian organizations implement the LGPD principles, but existing tools to support the elicitation of privacy requirements still need to be used. Furthermore, software requirements and construction are the areas of knowledge most impacted by the LGPD, and most agile teams use user stories to elicit privacy requirements. Although agile teams and Brazilian organizations are more concerned with user data privacy issues after the LGPD became effective, they still face challenges in privacy requirements elicitation [5].

The study by Raza [6] titled “Managing ethical requirements elicitation of complex socio-technical systems with critical systems thinking: A case of course-timetabling project” discusses the development of complex socio-technical systems with an emphasis on ethics and the perspectives of marginalized groups. The writers employed critical systems thinking to examine moral dilemmas and opposing opinions to create an equitable system for everyone. The paper utilized a case study of a scheduling issue at a university to show how critical systems thinking may be employed. Future complex information system designs may benefit from using the methodology utilized in this study [6].

A study of decision-making techniques for requirement engineering in agile software development is presented in the paper “Systematic Literature Review on Decision-Making of Requirement Engineering from Agile Software Development” by Ghozali et al. The authors examined eight studies from 2017 to now, identifying 11 decision-making approaches and seven difficulties. The study offers perceptions into requirements engineering and agile software development decision-making. Nevertheless, the findings’ validity might be limited by the scant number of publications examined and the stringent quality evaluation standards utilized [7].

In [8], Pathak et al. address the challenges of extracting sentiment related to specific topics from social media data using deep learning techniques. The study highlights the high-speed nature of data generation on social media platforms and the necessity for timely decision-making based on user sentiments. To address this, the authors developed a model employing online latent semantic indexing with regularization to detect topics in sentence-level streaming data, coupled with a topic-level attention mechanism within a long short-term memory (LSTM) network for sentiment analysis. This paper has many limitations, such as the model’s focus being on text-only modality; it does not incorporate multi-modal data, which could provide richer contextual insights, particularly in a social media environment where images, videos, and links often accompany text. Also, the research primarily relies on supervised learning methods, necessitating extensive labeled datasets that may not always be feasible or available for different contexts or less popular topics. The paper does not extensively explore the handling of ambiguities and the subtleties of language, such as irony or sarcasm, which are prevalent in social media texts and can skew sentiment analysis results. The dependency on specific datasets and the lack of generalization tests across diverse social media platforms limit the broader applicability of the findings.

In [9], Li et al. present a method for fine-grained sentiment analysis of Chinese social media content, explicitly focusing on humor detection using Internet slang and emojis. The technique employs an attention-based bi-directional long short-term memory (AttBiLSTM) neural network to analyze the sentiment of posts on Weibo, the largest Chinese social network. The study’s main limitations can be summarized in the following way: The method is tailored explicitly for Chinese social media

content, which may limit its applicability to other languages or cultural contexts without substantial modifications. Also, Internet slang and emoji meanings can evolve rapidly, which challenges updating the lexicons without frequent revisions. The model's performance heavily relies on the availability of accurately labeled training data, which can be resource-intensive to gather and update.

In [10], Chen et al. delve into sentiment analysis of social media content using deep learning technologies. This study uses data from Taiwan's most significant online forum, specifically the Militarylife PTT board, to develop and test a sentiment analysis framework. The research aimed to enhance sentiment classification by creating a military-specific sentiment dictionary and exploring the performance of different deep-learning models with varied parameter settings. The limitation can be summarized in the study focused only on a specific military board within a Taiwanese forum, which might limit the generalizability of the findings to other social media platforms or topics. The models and sentiment dictionary are tailored for Chinese language content, particularly with military jargon. Adapting the approach to different languages or contexts may require significant modifications. The study solely analyzed textual data, not considering other forms of media like images or videos often used in social media, which could also carry significant sentimental information.

In [11], Alsayat focuses on enhancing sentiment analysis in social media contexts. This is achieved through a novel ensemble deep learning model that leverages advanced word embedding techniques and a Long Short-Term Memory (LSTM) network. While the model shows promising results, it is primarily tested on COVID-19-related data and specific review datasets, which may limit its generalizability to other contexts or types of social media data. Combining several advanced techniques, the ensemble model might be computationally intensive and require significant resources, which could be a barrier for real-time applications. The reliance on FastText and other predefined word embeddings could limit the model's ability to adapt dynamically to new slang or terminology not present in the training data.

In [12], Kokab et al. explore the integration of sentiment analysis into various domains of social media security. The article highlights how sentiment analysis is applied to enhance security measures against threats like data deception, spam, and cyberattacks in online environments. The study notes that while sentiment analysis has made significant strides, it is often limited to text data, disregarding other forms of media content like images and videos accompanying social media posts. The effectiveness of sentiment analysis heavily relies on the quality and breadth of the dataset used, which can vary significantly across social media platforms. The paper also hints at the ethical and privacy concerns related to analyzing social media data, where the line between security measures and privacy invasion can become blurred.

In [13], Obaidi et al. systematically review the use of sentiment analysis tools in investigating sentiment analysis tools applied to SE (software engineering), focusing on different application scenarios, data sources, methodologies, and common challenges. A systematic literature review was conducted, analyzing data from 80 papers involving sentiment analysis tools in SE. The study examined application domains, purposes, datasets, approaches, and difficulties related to using these tools. The review mainly includes studies on open-source projects, potentially overlooking sentiment analysis applications in more proprietary or industrial contexts. The paper concludes by emphasizing the need for more domain-specific tools, better cross-platform consistency, and improved handling of linguistic nuances in sentiment analysis within software engineering.

Future research might incorporate more digital libraries and broaden its inclusion criteria to increase the study's validity. Some existing approaches to user requirements from social media are shown in [Table 1](#).

**Table 1:** Existing approaches of user requirement

#	Paper	Technique	Dataset	Results	Limitations
1.	[14]	Scrum scale model, coordination theory	Social media	Accuracy: 86%	Limited to scalability requirements
2.	[1]	Prototyping, brainstorming, mind mapping	Case study and survey	Accuracy: 88% F1-score: 86%	Limited scope
3.	[15]	Prototyping, hypothesis, and test	Questionnaire and interview	Precision: 90% Recall: 89%	Have biased opinions from irrelevant knowledge.
4.	[16]	Quantitative method and a survey for data collection	A survey was conducted among 34 Agile software practitioners in Jordan.	Accuracy: 89.35% F1-score: 88.25%	Smaller-scale research might need more reliable outcomes
5.	[17]	Agile delivery, data-driven framework, Meta-model integration	Microblogging	F1-score: 92.45%	Heterogeneous and unstructured data sources are challenging to integrate and process

From the above discussion, most existing requirement elicitation methods are heterogeneous and may require survey questionnaires or interviews to gather user requirements. All such techniques require human resources and plenty of time. Moreover, these methods have limited scope, potential biases, and small sample sizes, making these approaches less accurate. Therefore, there exists a need for careful consideration of these limitations to understand the applicability and potential constraints of each study's results in practical software development scenarios.

### 3 Proposed Methodology

The methodology of this research involves several stages, each designed to refine the process of eliciting user requirements from social media data through the use of advanced data processing and machine learning techniques:

#### 1. Data Collection:

- **Sources:** User comments were systematically collected from popular social media platforms, specifically Twitter and Facebook, using their respective APIs. The focus was capturing a broad spectrum of user opinions and feedback on software products and services.
- **Tools:** For Twitter, the Tweepy API was utilized to stream and capture live tweets, while for Facebook, where API access may be restricted, web scraping techniques were employed to extract comments from public posts and fan pages.

#### 2. Data Preprocessing:

- **Cleaning:** The raw data underwent a rigorous cleaning process to remove noise, such as irrelevant symbols, URLs, and user-specific identifiers. This step was crucial to enhance the quality of the text analysis.
- **Normalization:** Text data were normalized to lowercase, and stopwords were removed to ensure consistency in processing.



- **Tokenization and Lemmatization:** The cleaned texts were tokenized into words, and lemmatization was applied to reduce words to their base or root form.

### 3. Feature Extraction:

**Techniques:** Advanced Natural Language Processing (NLP) techniques were applied to extract meaningful features from the preprocessed text. The study used Graph-based Neural Networks (GNN) to manage the complex social media data structure effectively.

**Implementation:** The GNN was employed to analyze the interconnected data of user interactions and sentiments, allowing the model to capture contextual relationships between words in user comments.

### 4. Sentiment Analysis Model:

- **Model Development:** A deep learning-based sentiment analysis model was developed. This model used layers of neural networks to classify texts into different sentiments, which indicate user requirements.
- **Training:** The model was trained on a labeled dataset where user comments were manually annotated as requirements or non-requirements. This training involved adjusting the model parameters to minimize classification errors.
- **Classification and Analysis:**
- **Classification:** Post-training, the model classified new user comments of requirements and non-requirements based on the learned patterns.
- **Validation:** The model's effectiveness was validated through a series of tests that measured accuracy, precision, recall, and F1-score against manually annotated test data.

### 5. Evaluation and Refinement:

- **Performance Metrics:** The model performance was evaluated using standard metrics to ensure its accuracy and reliability in real-world scenarios.
- **Feedback Loop:** Feedback from the initial classification rounds was used to refine the model iteratively. This approach ensured continuous improvement in response to dynamic user interactions on social media.

Automatic Requirement Elicitation from Social Media (ARESM) is the process of obtaining user input from social media platforms and deriving needs from that feedback. The ARESM method flowchart is shown in [Fig. 1](#), which includes five essential steps: gathering user feedback, cleansing feedback data, categorizing needs, calculating text similarity, and visualizing the outcomes.

#### 3.1 User Feedback Review Collection

Users' comments from Facebook and Twitter, two social media sites, are collated for this study. Due to the complex nature of open text, as shown in [Fig. 2](#), the data collection process is quite challenging.

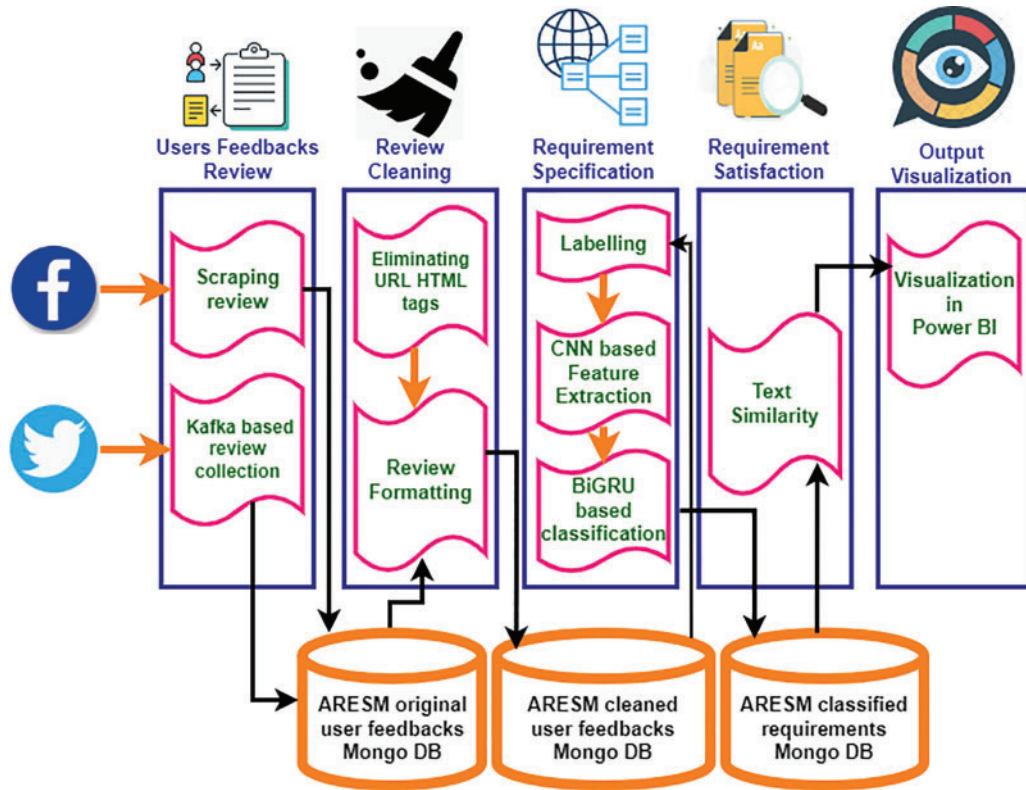


Figure 1: Proposed sentiment analytic model for automatic requirement elicitation

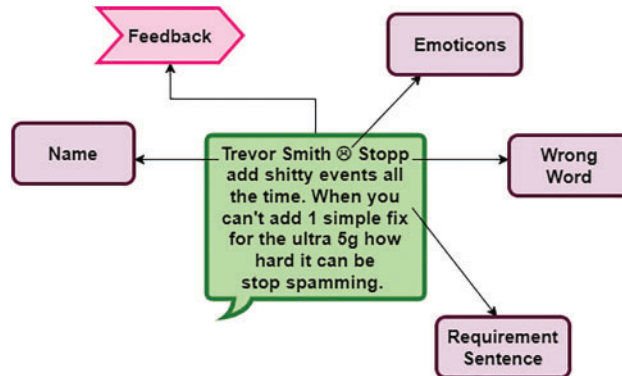
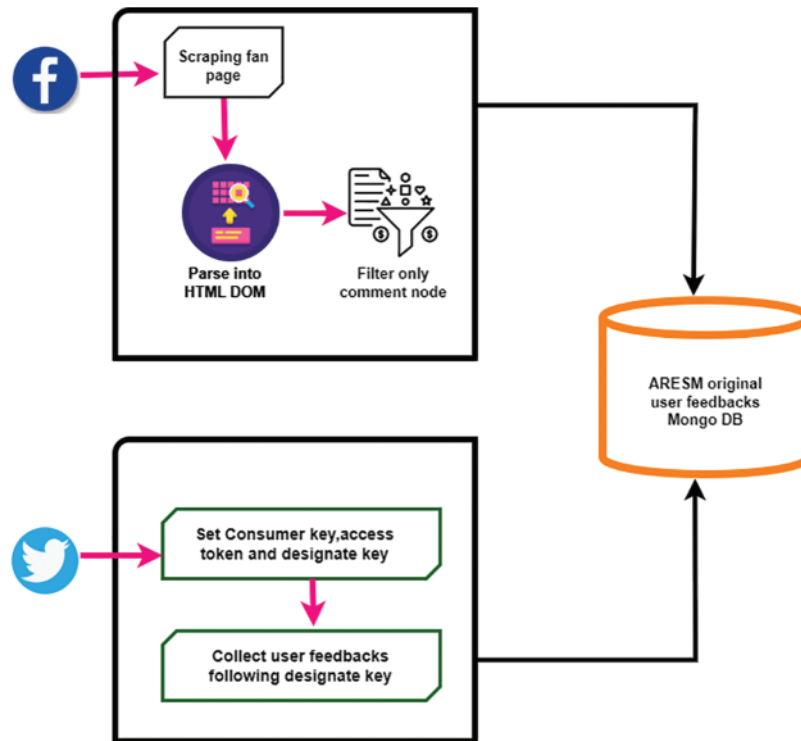


Figure 2: The structure of a facebook review

Fig. 3 shows how the initial user feedback Mongo DB of ARESM stores user feedback from Facebook and Twitter. The Kafka tool is used with the Twitter API to collect user feedback from Twitter. On the other hand, web scraping techniques are used mainly to extract user feedback from Facebook fan pages and gather user feedback from Facebook.





**Figure 3:** Collection of data from social media user feedback

During the data collection process, a Twitter developer account was built with the consumer key, consumer secret, access token, and access token secret via the Kafka tool. The stream Listener method of the Tweepy library uses these credentials to get user input from Twitter. However, this study concentrates on two terms, “Pokemon” and “Pokemon Go,” to collect user input. Due to privacy regulations, Facebook does not provide open APIs. Thus, web scraping is used to gather user input from the Pokemon Go Facebook fan page. As soon as the information has been scraped, it is arranged in the HTML (Hypertext Markup Language) DOM (Document Object Model) format and changed into a tree structure, where each node denotes an item shown in Fig. 4.

Different sorts of nodes, including headers, body text, personal information, cover page components, menu items, page-specific data, individual posts, and comments, are included in the HTML DOM document. In this study, ARESM only gathers data from the comment nodes, which house user input. These comments are potential new software specifications or ideas for improving already-existing functionalities.

### 3.2 User Feedback Cleaning

After the successful data collection, each commenter’s first and last names are eliminated to avoid bias in the work. It might not be easy to understand feedback from social media sites due to URLs (Uniform Resource Locators) and slang words, which need a cleaning process. Fig. 5 shows the cleaning process workflow that applied filters to handle incorrect language, emojis, and out-of-context terminology. The main goal of this method is to keep essential vocabulary linked to requirements while removing superfluous words from each user input.

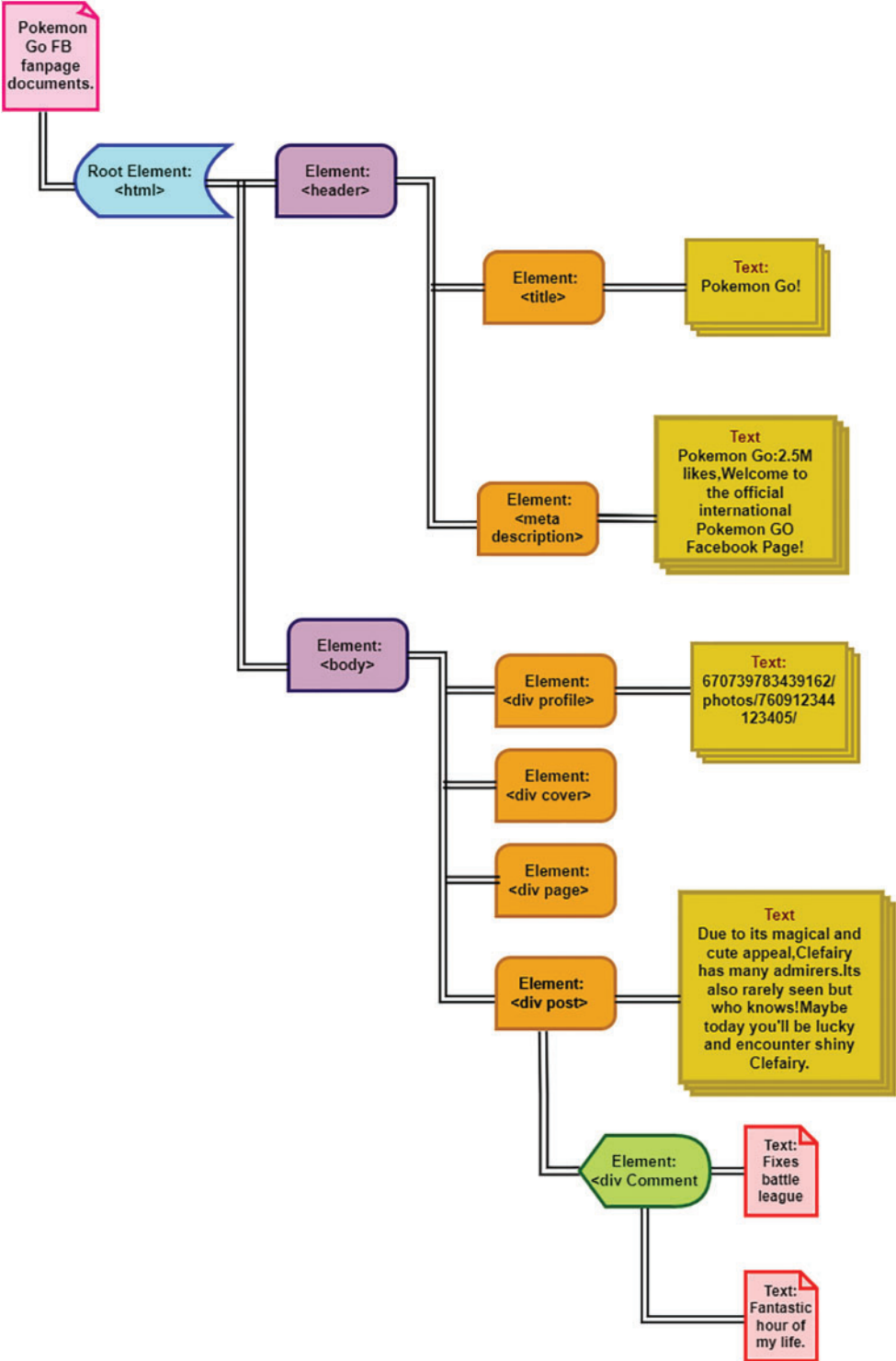
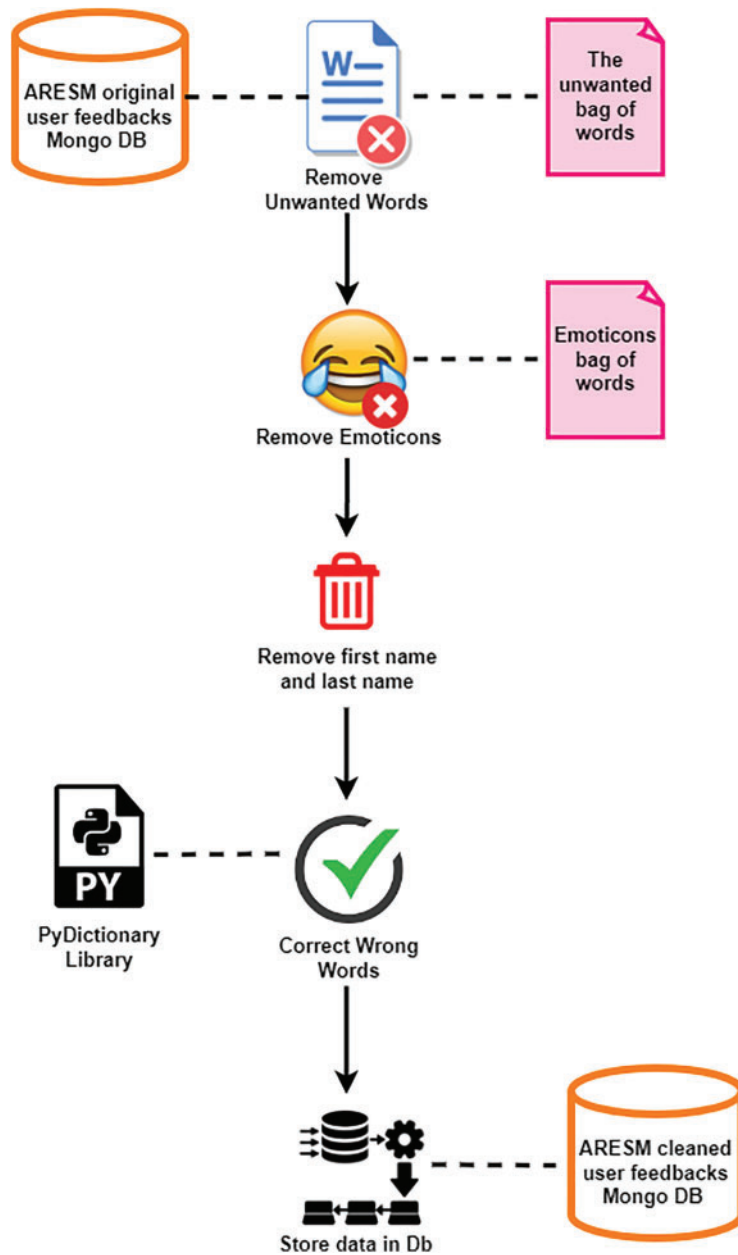


Figure 4: Tree formation architecture reviews



**Figure 5:** The cleaning process from User comments

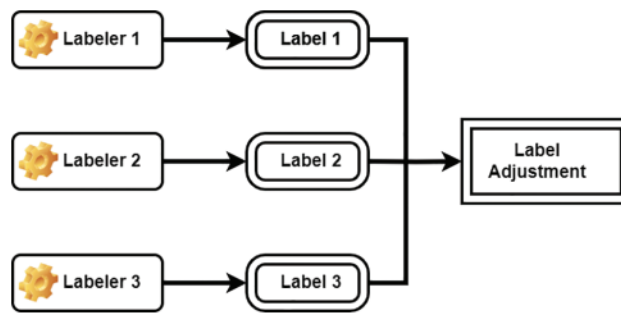
In addition to the cleaning process, the Bag-of-Words (BOW) and emoticon BOW have been used to remove unwanted words and to add more valuable non-contextual keywords and emoticons, respectively, by comparing user feedback to the offensive words BOW and emoticon BOW, ARESM (Automatic Removal of Extraneous Sentences Module) efficiently purges and filters user comments. The PyDictionary Library is used to rectify incorrect words. Fig. 5 shows how the cleaned user feedback is kept in the ARESM Cleaned User Feedbacks MongoDB.

### 3.3 Requirements Classification

The requirement classification phase classifies each sanitized user input as a necessity or a non-requirement. The critical steps of this phase are further divided into labeling, feature extraction, and classification.

#### 3.3.1 Labeling

Due to the rare availability of labeled datasets, the collected dataset has been labeled manually. To improve accuracy and minimize bias in the labeling, the cleaned user input goes through a labeling procedure with three labelers. These labelers attend a face-to-face session to obtain instructions on the labeling procedure. As shown in Fig. 6, each labeler labels the identical collection of user feedback from the cleaned user feedback Mongo DB. Each user’s comment is assigned a label based on a voting process. A user comment is classified as a need if two out of three labelers do so. On the other hand, a user feedback item is classified as such if two out of three labelers classify it as such. Examples of human labeling are shown in Table 2, along with a sample of user input that communicates a need. In this case, the comment implies that the Go Plus gadget needs to be fixed, identifies a problem with the most recent update, and expresses displeasure that it cannot be used despite paying for it.



**Figure 6:** Three labeler adjustment model for manual labelling process

**Table 2:** Human-based labeling for requirement elicitation

Requirements	Non-requirements
I require the Go Plus to be repaired, as it has not been functional since the update. I have invested a sufficient amount in its usage, and now it seems you have made changes preventing me from utilizing it. I request your immediate attention to resolve this issue.	Throughout the two-hour duration spent walking around the shopping center parking lot, my companion failed to procure any items for me.
I am currently experiencing compatibility issues between my S20 device and Pokémon Go. Unfortunately, no remedies, updates, or responses have been provided to address this matter. I am utilizing the hashtag #pokfixS20 to bring attention to this issue.	I have obtained all the items except the Chalky stone and Tropical shell.

(Continued)

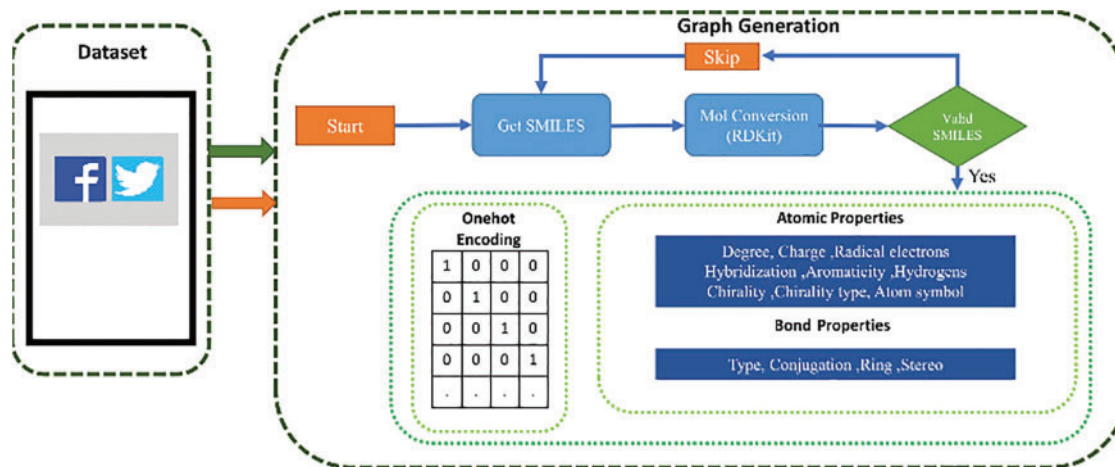
**Table 2 (continued)**

Requirements	Non-requirements
I have encountered an issue where the game consistently crashes during spawning, wasting time and effort without any satisfactory outcome. The current state of affairs reflects an unsatisfactory outcome, resembling the initial conditions I encountered when I first began.	I rarely utilize my buddy feature in the game, as I frequently find myself perplexed and mistaking it for a new Pokémon.  I am amazed by the abundance of remarkable gifts present here. However, it is unfortunate that the only items I seem to be fixated on are the crack rock and mushroom.

The submitted feedback, “In the two hours we spent roaming around the shopping center parking lot, my friend did not bring me anything.” does not qualify as a need since it is devoid of terminology related to requirements and does not point out a demand for resolving current problems or seeking new features.

### 3.3.2 Feature Extraction

Various machine learning techniques exist, such as Term Frequency Inverse Document Frequency of records (TF-IDF) and support vector machines for feature extraction. Still, deep learning models are better because they can automatically uncover pertinent patterns and representations from large amounts of raw text data, particularly in social media text and reviews. Graph-based Neural Networks (GNN), as shown in Fig. 7, can be a powerful way to extract features from reviews, primarily when the thoughts are related to a graph structure, such as user interactions, links between users and requirements, or any other type of interconnected data. GNNs may capture the intrinsic links and dependencies between graph parts, providing essential context for feature extraction from social media evaluations.



**Figure 7:** GNN architecture for feature extraction

During this process, the social media review data are initially displayed as a graph, with nodes denoting reviews, users, requirements, or any other relevant elements and edges denoting links between these entities (such as user interactions, mentions, or co-occurrences). Future extraction requires meaningful representations that are further assigned to every node (such as reviews, users, etc.) and edge within the graph. These representations encompass text embeddings (utilizing pre-trained word embeddings), metadata attributes, or other pertinent data. When dealing with reviews, the textual content can be encoded using methodologies like Word2Vec, GloVe, or BERT (Bidirectional Encoder Representations from Transformers). In each layer of the GNN, nodes gather information from their neighbors to update their features. This process involves aggregating data from related studies, users, or products when applied to reviews. Through this, the GNN learns to capture interactions and influences between different elements within the graph.

Fig. 7 describes the architecture of a Graph-based Neural Network (GNN) used for feature extraction. The GNN architecture consists of five steps as follows:

**1. Graph Representation:**

- The social media data, including reviews, user interactions, and other relevant elements, are structured into a graph. In this graph, nodes represent user comments, reviews, users, and requirements, while edges represent relationships like user interactions, mentions, or co-occurrences.

**2. Feature Extraction Layer:**

- Each node and edge in the graph is assigned meaningful representations. These representations include text embeddings (utilized from pre-trained models like Word2Vec, GloVe, or BERT), metadata, and other relevant data.
- The feature extraction process at each node involves gathering and aggregating information from neighboring nodes, which could be related users, reviews, or sentiments. This aggregation helps update each node's features based on the surrounding context.

**3. Message Passing Layers:**

- The GNN processes the graph through multiple message-passing layers. In these layers, nodes update their states based on the information received from connected nodes. This helps capture the contextual relationships and dependencies among various parts of the graph.

**4. Attention Mechanisms and Pooling:**

- Attention mechanisms are used to emphasize essential nodes or edges during the message passing, which is particularly useful when specific users or themes significantly influence the overall sentiment or aspects of the discussion.
- Graph pooling techniques might reduce the graph's complexity while retaining essential features. This is particularly useful in large and complex graphs, as it helps summarize information at different levels of granularity.

**5. Output for Downstream Tasks:**

- The final node representations, refined through multiple processing layers, become valuable for downstream tasks like sentiment analysis or aspect extraction. These tasks might involve classifying reviews as positive or negative, predicting sentiment scores, or extracting critical aspects mentioned in the reviews.



The GNN architecture leverages the inherent graph structure of social media data to perform deep feature extraction, which is crucial for accurately analyzing and interpreting user sentiments and requirements.

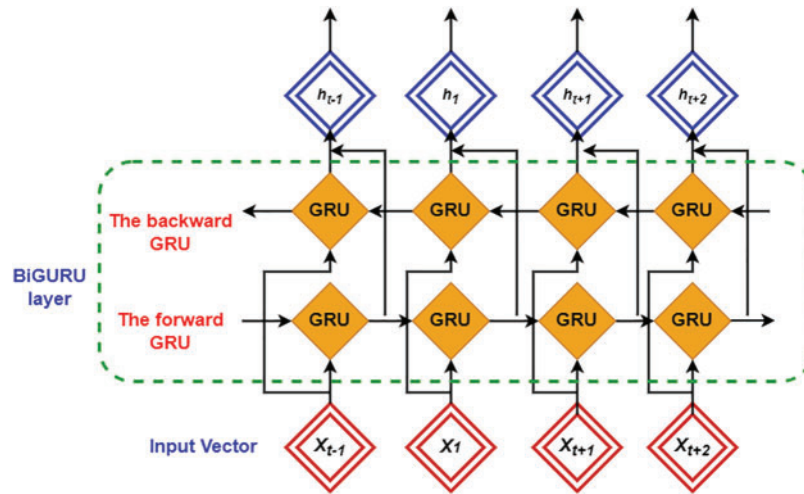
After the GNN processes the graph data through multiple message-passing layers, the node representations become valuable for downstream tasks like sentiment analysis or aspect extraction in social media reviews. These representations could be utilized to predict sentiment scores, classify reviews as positive or negative, or extract prominent aspects mentioned in the reviews. The attention mechanisms help emphasize important nodes or edges during message passing. This can be particularly useful when certain studies or users strongly influence sentiment or aspects more than others. In cases where the graph is large and complex, graph pooling techniques can summarize information at different levels of granularity. Graph pooling can reduce the computational complexity while retaining essential features from the graph. After the successful GNN process, some user feedback and extracted features are shown in [Table 3](#).

**Table 3:** User feedback along with extracted features using GNN

S. No.	Feedback	Feature extraction	Remarks
1	Go Plus must be repaired. It has yet to work since the upgrade. I paid enough for its use, and now you've done something I can't! Please resolve the problem.	Need, to, fix, Go Plus, not worked, since	Polarity factor, Subjective factor, No. of requirements
2	I hope they will be helpful in the future . . . Maybe it is a Requirement for mega-evolution.	I hope they will be helpful in the future	Polarity factor, Subjective factor, No. of requirements
3	Our buddy should pick up helpful stuff, like items used to evolve specific Pokemon.	Our buddy should pick up helpful staff	Polarity factor, Subjective factor, No. of requirements

### 3.3.3 Classification

This sub-process aims to classify each user's feedback as a necessity or a non-requirement. Labeled user input and the collected features are used to train the model. In this work, a Gated Recurrent Unit (GRU) based classification model, as shown in [Fig. 8](#), has been proposed to categorize each review into requirement and non-requirement elicitation. The key reason for using GRUs is the presence of sequential patterns and dependencies in text data.



**Figure 8:** The working mechanism of the GRU model for classification

The embedding layer of GRU converts the input features into dense vector representations using word embeddings. In contrast, the substantial layers capture sequential information from the review that can be adjusted into several hidden units in the GRU, which controls the model's capacity to learn complex patterns. The fully connected layer transforms the learned features into a format suitable for classification. The output layer, usually called the softmax layer, produces class probabilities (requirement vs. non-requirement). After the formation of GRU, it has been training on labeled data. A built-in loss function based on binary cross-entropy has been adopted for the final classification.

#### 4 Analysis and Results

The effectiveness of the proposed ARESM method for requirement elicitation is evaluated through various performance metrics, including precision, recall, F-score, and accuracy. These metrics highlight the method's ability to accurately categorize user feedback and derive meaningful needs from social media input.

##### 4.1 Evaluation Parameters

Accuracy, precision, and recall are the three classification metrics used to assess the performance of the proposed framework. Each of these metrics' formulations is provided in Eqs. (1)–(3), respectively. These measurements are crucial benchmarks for evaluating the model's efficiency and performance.

$$\text{Accuracy} = \frac{\text{sum}(\text{True Positives}, \text{True Negatives})}{\text{sum}(\text{True Negatives}, \text{False Positives}, \text{True Positives}, \text{False Negatives})} \quad (1)$$

$$\text{Precision} = \frac{\text{count}(\text{True Positives})}{\text{sum}(\text{True Positives}, \text{False Positives})} \quad (2)$$

$$\text{Recall} = \frac{\text{count}(\text{True Positives})}{\text{sum}(\text{True Positives}, \text{False Negatives})} \quad (3)$$

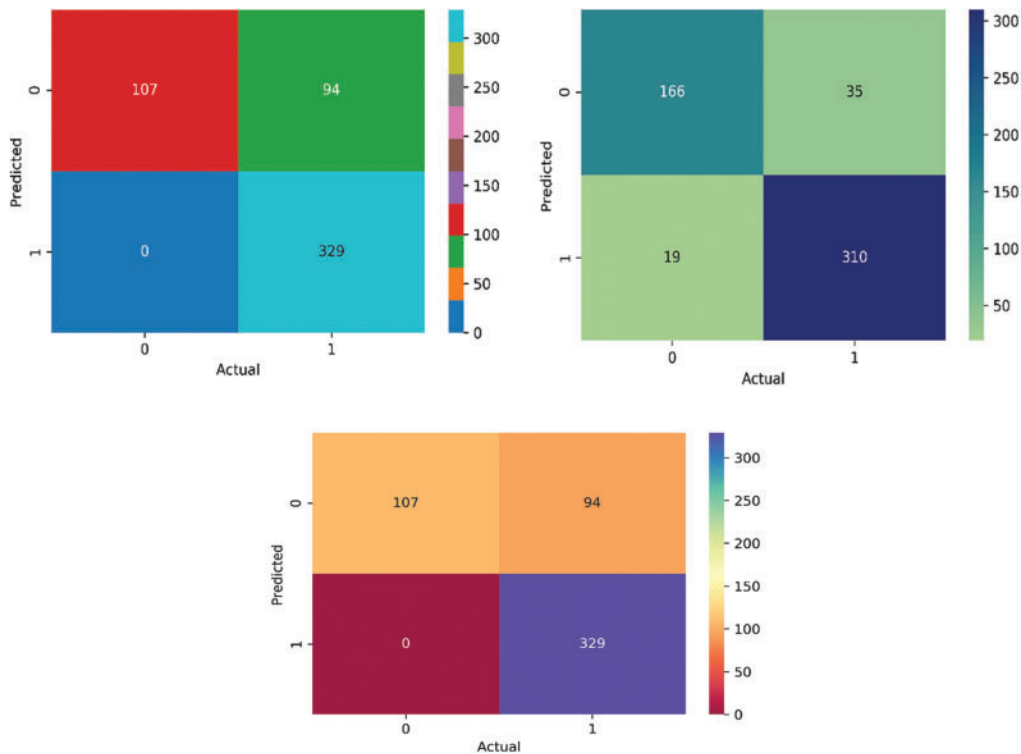
### 4.2 Baseline Method

The following baseline models have been considered to evaluate the performance of the proposed model:

- **Nayebi et al. [18]**, utilizing Naive Bayes Machine Learning, gather and categorize user reviews from Twitter, the Play Store, and the App Store.
- **Guzman et al. [19]** put forth a method for the automated categorization, clustering, and prioritization of tweets during software evolution from Twitter, employing the Multinomial Naïve Bayes technique.
- **Sawhney et al. [20]** and colleagues employ three deep learning models, namely Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Contextual Long Short-Term Memory (CLSTM) methods, to extract sentences from Twitter.

### 4.3 Results

The experimental results are presented graphically in Fig. 9, illustrating the performance of the proposed ARESM approach across different categories (such as Needs and Non-Needs) in terms of confusion matrix on various datasets. The results clearly demonstrate that the proposed work correctly predicts the true and false values of each class. Fig. 9 presents the performance of the ARESM model in terms of a confusion matrix for different datasets. It visually illustrates how the model predicts true and false values for each class across the DS-I, DS-II, and DS-III datasets. The matrix helps understand the true positives, true negatives, false positives, and false negatives obtained by the model, thereby showcasing its classification ability for different categories such as ‘Needs’ and ‘Non-Needs’.



**Figure 9:** The performance matrix of ARESM on DS-I, DS-II and DS-III

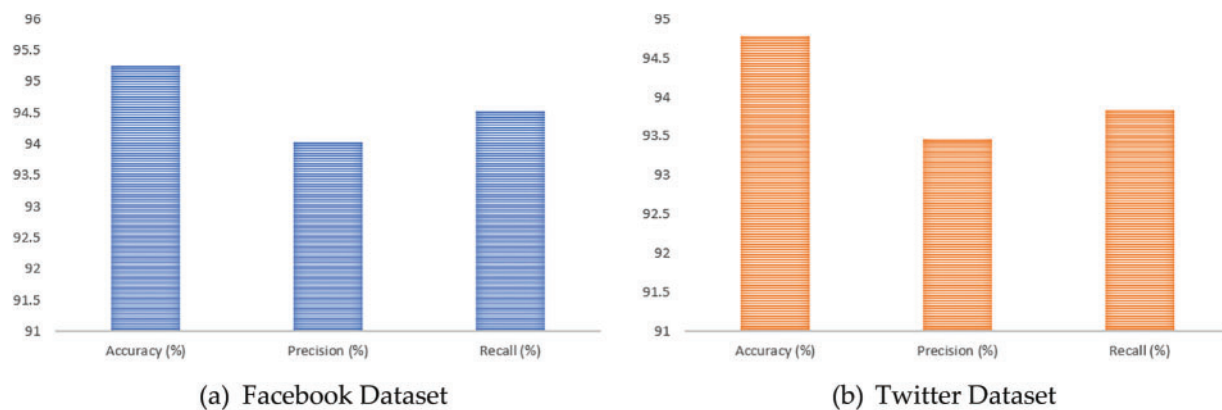
### Datasets DS-I, DS-II, and DS-III Explanation

- These datasets represent different collections of social media data used to test the ARESM model. The nature of the data within these datasets, such as the specific platforms or characteristics they include, directly impacts the testing and results of the model's performance. The segmentation into different datasets likely represents variations in data sources (e.g., various social media platforms or user demographics), which helps evaluate the model's robustness and effectiveness across varied inputs.

### Relation of Datasets to Research Questions

- The datasets DS-I, DS-II, and DS-III are instrumental in answering the research questions posed by the study, which are likely centered around the effectiveness of the ARESM model in accurately classifying user comments as requirements or non-requirements. By applying the model to different datasets, the research can assess its generalizability and precision across various types of social media data, thereby providing insights into its practical applicability and limitations.

Fig. 10 shows the performance of the proposed technique in terms of precision-recall and accuracy. The graphical visualization shows the proposed ARESM technique's effectiveness, which becomes evident when applied to the extensive Facebook dataset. In this context, its accuracy rate reaches an impressive 95.26%. This accuracy metric underscores the technique's remarkable capability to discern positive instances among the cases predicted as positive correctly. This accomplishment is critical as it signifies the technique's precision in identifying significant value cases. Furthermore, the technique's recall rate on the Facebook dataset is 94.53%. This metric represents the proportion of positive cases accurately identified among all the actual positive instances present within the dataset.

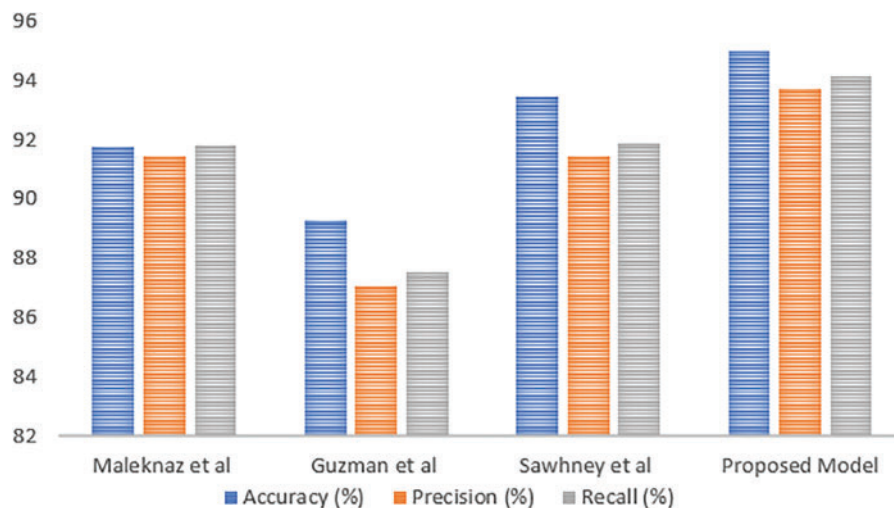


**Figure 10:** The percentage-wise performance of ARESM

It highlights the technique's sensitivity to capturing positive instances accurately. The precision score, calculated at 94.74%, comprehensively evaluates the technique's effectiveness in accurately categorizing positive instances. This metric considers the actual positive instances concerning both true and false positives, thus providing a nuanced understanding of the model's classification precision. Additionally, the ARESM technique extends its prowess to the realm of the Twitter dataset, achieving an accuracy rate of 94.78%. This metric denotes the proportion of positive and negative instances the technique correctly classifies out of the total cases evaluated from the Twitter dataset. The technique's proficiency across different datasets reaffirms its versatility and capacity for accurate classification.

Comparing the ARESM technique with a baseline method further underscores its superiority. The proposed model consistently outperforms the baseline across a range of metrics. The technique accurately categorizes instances as positive, with a precision score of 93.75%. The recall metric, quantified at 94.16%, reflects the technique’s effectiveness in accurately identifying positive instances among all the actual positives. Moreover, the model achieves an accuracy rate of 95.02%, further solidifying its capacity for precise classification.

It’s important to note that even though the differences in metrics between the proposed ARESM method and the baseline are relatively modest, the consistent superiority of the ARESM approach highlights its potential for achieving more accurate requirement elicitation from social media feedback. This potential is visually depicted in Fig. 11, where the comparison between the two methods is visually presented. The results support the notion that the ARESM technique possesses the qualities necessary to enhance the accuracy of requirement elicitation from social media inputs. Fig. 11 illustrates the comparative performance of the proposed ARESM model against various baseline approaches in sentiment analysis and requirement elicitation from social media data. The figure presents accuracy, precision, recall, and F1-score metrics to demonstrate how ARESM outperforms other methods. Each approach is evaluated on the same metrics, highlighting ARESM’s consistent superiority in correctly identifying and categorizing user requirements from social media feedback. The datasets used in these comparisons are from social media platforms like Facebook and Twitter. These platforms provide a rich source of user comments, which are leveraged to test the effectiveness of the proposed model against the baselines. The data used in these comparisons are primarily newly collected through systematic methods using APIs for Twitter and scraping for Facebook. This approach ensures a fresh dataset that is relevant and specific to the study’s needs, focusing on current user opinions and feedback related to software products and services. This detailed comparison and the nature of the datasets are crucial for validating the robustness and applicability of the ARESM model in real-world scenarios, ensuring that the model not only performs well in controlled tests but also effectively handles actual user data from social media.



**Figure 11:** Comparison of proposed model with baseline approaches

- *Respond to Research Question 1*

Facebook has 381 non-criteria in addition to 153 requirements. Twitter has 178 non-requirements as opposed to 40 requirements. There are a total of 600 user comments. Four hundred eight user feedbacks are not required, whereas 192 are. The Facebook and Twitter media platform datasets demand densities of 43.12%, 21.44%, and 32.17%, respectively. The analysis regarding how density requirements appear on Facebook and Twitter in the context of the research question focuses on assessing the amount and type of user feedback that qualifies as requirements *vs.* non-requirements across these platforms. Understanding these densities involves comparing the counts categorized as requirements and non-requirements based on user feedback collected from both platforms. **Facebook** shows a higher density of requirements compared to Twitter. This is quantified with specific numbers indicating a larger volume of requirements identified in the Facebook dataset. This might be due to the nature of interactions on Facebook, which potentially allows more detailed user feedback due to its format and user engagement patterns. **Twitter**, on the other hand, has a lower requirement density. The nature of Twitter, with its character limit and rapid interaction style, may contribute to less detailed feedback that qualifies as a requirement.

The densities of requirements on each platform are quantitatively depicted, directly answering how requirement densities manifest on Facebook *vs.* Twitter. These metrics are crucial as they indicate the volume and quality of data that can be extracted from each platform concerning user requirements.

Furthermore, these results pertain to data collected explicitly for this study from the platforms mentioned rather than using publicly available datasets. This tailored collection ensures that the data is relevant to the specific research questions posed, mainly focusing on how user requirements are expressed in social media contexts related to software products and services

- *Respond to Research Question 2*

The classification of user feedback in terms of needs or no requirements using GRU. Are respondents by 600 users. One hundred sixty-two user feedbacks have been tested once 500 user feedbacks have been learned. The confusion matrix is displayed in [Table 4](#).

**Table 4:** Requirements density for RQ1 (*Research Question 1*)

	Requirements	Non functional requirements	Total	Requirements density
Facebook	156	233	386	43.12%
Twitter	43	181	220	21.44%
Total	196	411	603	35.22%

[Table 5](#) presents a confusion matrix detailing the classification outcomes of user feedback using the Gated Recurrent Unit (GRU) model, aimed at distinguishing between requirements and non-requirements. The matrix highlights the true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values. Specifically, the model correctly identified 33 instances as requirements (TP) and 38 as non-requirements (TN), while 31 non-requirements were incorrectly classified as requirements (FP), and 10 requirements were misclassified as non-requirements (FN). These results underscore the model's capability in accurately categorizing user feedback, though some misclassifications still occur, indicating areas for potential improvement in the model's precision and recall.



**Table 5:** Confusion matrix

Label/predict	Requirement (true)	Non functional requirements (false)
Requirement	T-P (33)	F-P (31)
Non requirements	F-N (10)	T-N (38)

**Table 6:** The accuracy of three label

	Accuracy	Precision	Recall	F-Measure
Three label	65%	51.72%	81.08%	63.15%

In [Table 6](#), the accuracy is displayed. Three people label the user comments. Each labeler labels identical user input as requirements or non-requirements. The Naive Bayes machine learning model is then trained and evaluated using tagged user feedback from each labeler. Accuracy for three labelers is 65%, precision is 51.72 %, recall is 81.08%, and F-Measure is 63.15%. The advancements provided by these algorithms are significant as they allow for more dynamic and responsive software development practices. Integrating user feedback directly from social media into the development process allows the software to continuously improve and meet user expectations. Additionally, the ability to operate across different languages and cultural contexts increases the global applicability of these techniques, potentially transforming how user requirements are gathered and analyzed on a large scale. Significant Advancements Over Existing Methods can be summarized as follows:

- Unlike traditional methods, which are often manual and time-consuming, the proposed model automates user feedback collection from social media platforms using APIs. This method is faster and captures real-time user sentiments, which are crucial for timely software updates.
- Using Graph-based Neural Networks (GNN) for feature extraction represents a significant advancement. GNNs effectively manage the complex structure of social media data, allowing the model to capture deeper contextual relationships between words in user comments, which traditional methods might not leverage.
- The proposed model demonstrates superior performance metrics, notably in accuracy and precision. For example, the model achieves an impressive accuracy rate of 95.02%, which significantly surpasses existing benchmark methods. This high level of accuracy ensures that the software development process can be more responsive to actual user needs by accurately classifying and responding to user requirements.
- The model incorporates sentiment analysis to differentiate between requirements and non-requirements in user comments. This approach allows for a more nuanced understanding of user feedback, distinguishing between mere opinions and actionable requirements.
- The model utilizes feedback loops to refine its predictions continuously. This iterative refinement helps the model stay relevant and accurate as it adapts to new user feedback and evolving language use on social media platforms.
- The proposed methodology is designed to be adaptable across different domains and languages, which enhances its utility for global software development practices. This contrasts with many traditional methods that are often context-specific.

Overall, the proposed algorithms improve the efficiency and effectiveness of requirement elicitation and contribute to a more user-centered software development approach, which is essential in today's fast-paced digital world.

## 5 Conclusion

User requirements are of ultimate need when there is a need to perform any changes within software systems. In this regard, software requirements are gathered by following traditional methods of conducting interviews, distributing questionnaires, or coordinating Joint Application Development (JAD) meetings that are pretty hectic and time-consuming. An automatic sentiment analysis-based user requirement elicitation model has been proposed in this research work. The proposed model leverages appropriate APIs to collect user reviews from social media platforms. These collected reviews undergo preprocessing, including filtering for feature extraction, and are ultimately classified as either requirements or non-requirements. The experimental results and evaluations have demonstrated the superiority of this proposed model over existing benchmark methods. With an impressive accuracy rate of 95.02%, the model showcases its potential to significantly enhance the software development process by incorporating online user feedback. In future work, we aim to further enhance the proposed model by exploring advanced natural language processing techniques, such as sentiment analysis and context-aware classification. Additionally, we plan to incorporate a temporal dimension to capture evolving user preferences and trends over time, enabling more accurate software adaptation. Furthermore, extending the model's applicability to a more comprehensive range of domains and languages will be crucial to its broader adoption and impact on global software development practices.

**Acknowledgement:** The authors would like to acknowledge the support of Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Funding Statement:** This research is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Author Contributions:** All authors contributed equally. The authors confirm contribution to the paper as follows: study conception and design: Deema Mohammed Alsekait, Asif Nawaz; data collection: Mehwish Bukhari; analysis and interpretation of results: Deema Mohammed Alsekait, Daa Salama AbdElminaam; draft manuscript preparation: Asif Nawaz, Ayman Nabil; supervision, methodology, conceptualization, Formal analysis, writing-review & editing: Daa Salama Abd Elminaam; methodology, formal analysis, writing-review & editing. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data is available with the corresponding author and can be shared on request.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] A. Aldave, J. M. Vara, D. Granada, and E. Marcos, “Leveraging creativity in requirements elicitation within agile software development: A systematic literature review,” *J. Syst. Softw.*, vol. 157, no. 3, pp. 110396, 2019. doi: [10.1016/j.jss.2019.110396](https://doi.org/10.1016/j.jss.2019.110396).
- [2] Y. Wang and L. Zhao, “Eliciting user requirements for e-collaboration systems: A proposal for a multi-perspective modeling approach,” *Requirements Eng.*, vol. 24, no. 2, pp. 205–229, 2019. doi: [10.1007/s00766-017-0285-7](https://doi.org/10.1007/s00766-017-0285-7).
- [3] E. D. Canedo, I. N. Bandeira, A. T. S. Calazans, P. H. T. Costa, E. C. R. Cançado and R. Bonifácio, “Privacy requirements elicitation: A systematic literature review and perception analysis of IT practitioners,” *Requirements Eng.*, vol. 28, no. 2, pp. 177–194, 2023. doi: [10.1007/s00766-022-00382-8](https://doi.org/10.1007/s00766-022-00382-8).
- [4] C. Pacheco, I. Garcia, J. A. Calvo-Manzano, and M. Reyes, “Measuring and improving software requirements elicitation in a small-sized software organization: A lightweight implementation of ISO/IEC/IEEE 15939: 2017—systems and software engineering—measurement process,” *Requirements Eng.*, vol. 28, no. 2, pp. 257–281, 2023. doi: [10.1007/s00766-022-00394-4](https://doi.org/10.1007/s00766-022-00394-4).
- [5] E. D. Canedo, A. T. S. Calazans, I. N. Bandeira, P. H. T. Costa, and E. T. S. Masson, “Guidelines adopted by agile teams in privacy requirements elicitation after the Brazilian general data protection law (LGPD) implementation,” *Requirements Eng.*, vol. 27, no. 4, pp. 545–567, 2022. doi: [10.1007/s00766-022-00391-7](https://doi.org/10.1007/s00766-022-00391-7).
- [6] S. A. Raza, “Managing ethical requirements elicitation of complex socio-technical systems with critical systems thinking: A case of course-timetabling project,” *Technol. Soc.*, vol. 66, no. 1, pp. 101626, 2021. doi: [10.1016/j.techsoc.2021.101626](https://doi.org/10.1016/j.techsoc.2021.101626).
- [7] R. P. Ghozali, H. Saputra, M. A. Nuriawan, D. N. Utama, and A. Nugroho, “Systematic literature review on decision-making of requirement engineering from agile software development,” *Procedia Comput. Sci.*, vol. 157, no. 5, pp. 274–281, 2019. doi: [10.1016/j.procs.2019.08.167](https://doi.org/10.1016/j.procs.2019.08.167).
- [8] A. R. Pathak, M. Pandey, and S. Rautaray, “Topic-level sentiment analysis of social media data using deep learning,” *Appl. Soft Comput.*, vol. 108, no. 6, pp. 107440, 2021. doi: [10.1016/j.asoc.2021.107440](https://doi.org/10.1016/j.asoc.2021.107440).
- [9] D. Li, R. Rzepka, M. Ptaszynski, and K. Araki, “HEMOS: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media,” *Inform. Process. Manag.*, vol. 57, no. 6, pp. 102290, 2020. doi: [10.1016/j.ipm.2020.102290](https://doi.org/10.1016/j.ipm.2020.102290).
- [10] L. C. Chen, C. M. Lee, and M. Y. Chen, “Exploration of social media for sentiment analysis using deep learning,” *Soft Comput.*, vol. 24, no. 11, pp. 8187–8197, 2020. doi: [10.1007/s00500-019-04402-8](https://doi.org/10.1007/s00500-019-04402-8).
- [11] A. Alsayat, “Improving sentiment analysis for social media applications using an ensemble deep learning language model,” *Arab J. Sci. Eng.*, vol. 47, no. 2, pp. 2499–2511, 2022. doi: [10.1007/s13369-021-06227-w](https://doi.org/10.1007/s13369-021-06227-w).
- [12] S. T. Kokab, S. Asghar, and S. Naz, “Transformer-based deep learning models for the sentiment analysis of social media data,” *Array*, vol. 14, no. 4, pp. 100157, 2022. doi: [10.1016/j.array.2022.100157](https://doi.org/10.1016/j.array.2022.100157).
- [13] M. Obaidi and J. Klünder, “Development and application of sentiment analysis tools in software engineering: A systematic literature review,” in *Proc. 25th Int. Conf. Eval. Assess. Softw. Eng.*, New York, NY, USA, 2021, pp. 80–89.
- [14] G. Brataas, A. Martini, G. K. Hanssen, and G. Ræder, “Agile elicitation of scalability requirements for open systems: A case study,” *J. Syst. Softw.*, vol. 182, no. 10, pp. 111064, 2021. doi: [10.1016/j.jss.2021.111064](https://doi.org/10.1016/j.jss.2021.111064).
- [15] H. F. Martins, A. C. de Oliveira Junior, E. D. Canedo, R. A. D. Kosloski, R. Á. Paldês and E. C. Oliveira, “Design thinking: Challenges for software requirements elicitation,” *Information*, vol. 10, no. 12, pp. 371, 2019. doi: [10.3390/info10120371](https://doi.org/10.3390/info10120371).
- [16] H. Saeeda, J. Dong, Y. Wang, and M. A. Abid, “A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project,” *J. Softw. Evol. Process.*, vol. 32, no. 7, pp. 1–24, 2020. doi: [10.1002/smr.2247](https://doi.org/10.1002/smr.2247).
- [17] A. Henriksson and J. Zdravkovic, “A data-driven framework for automated requirements elicitation from heterogeneous digital sources,” in *Proc. 13th IFIP Work. Conf., PoEM 2020*, Riga, Latvia, 2020, pp. 351–365.

- [18] M. Nayebi, H. Farrahi, G. Ruhe, and H. Cho, "App store mining is not enough," in *Proc. 2017 IEEE/ACM 39th IEEE Int. Conf. Softw. Eng. Companion*, Buenos Aires, Argentina, 2017, pp. 152–154.
- [19] E. Guzman, R. Alkadhi, and N. Seyff, "A needle in a haystack: What do twitter users say about software?," in *Proc. 2016 IEEE 24th Int. Requirements Eng. Conf. (RE)*, Beijing, China, 2016, pp. 96–105.
- [20] R. Sawhney, P. Manchanda, P. Mathur, R. Shah, and R. Singh, "Exploring and learning suicidal ideation connotations on social media with deep learning," in *Proc. 9th Workshop Comput. Approaches Subjectivity, Sentiment Soc. Med. Anal.*, Brussels, Belgium, 2018, pp. 167–175.