Research Article

Zeeshan Anwar*, Hammad Afzal, Ali Ahsan, Naima Iltaf, and Ayesha Maqbool

A novel hybrid CNN-LSTM approach for assessing StackOverflow post quality

https://doi.org/10.1515/jisys-2023-0057
received May 02, 2023; accepted October 17, 2023

Abstract: Maintaining the content quality on social media Q&A platforms is pivotal for user attraction and retention. Automating post quality assessment offers benefits such as reduced moderator workload, amplified community impact, enhanced expert user recognition, and importance to expert feedback. While existing approaches for post quality mainly employ binary classification, they often lack optimal feature selection. Our research introduces an automated system that categorizes features into textual, readability, format, and community dimensions. This system integrates 20 features belonging to the aforementioned categories, with a hybrid convolutional neural network–long short-term memory deep learning model for multi-class classification. Evaluation against baseline models and state-of-the-art methods demonstrates our system's superiority, achieving a remarkable 21–23% accuracy enhancement. Furthermore, our system produced better results in terms of other metrics such as precision, recall, and *F*1 score.

Keywords: community question answering, quality assessment, crowd sourcing, deep learning

1 Introduction

StackOverflow is the largest community question answering (CQA) site for programmers. About 30% of the questions posted on StackOverflow remain unanswered. The reason behind this is that the questioner is unable to attract experts to answer his question [1]. Content quality [2] is important to attract new users and retain existing users [3]. Moderators assess the quality of questions posted on StackOverflow and delete the low-quality questions. The daily volume of users' posts is quite high; therefore, assessing the quality of content manually is not practical. To overcome this, an automated system is required to determine the quality of content.

Generally, many answers are posted on CQA sites for a question but only a few answers are of worth [4]. All answers are not posted by domain experts, and the moderator can judge the quality of the answer, but it is not practical because it is based on the moderator's subjective assessment [3]. Assessing the quality of Q&As on CQA sites can be divided into three problems, i.e. determining the quality of questions [5], determining the quality of the answers [6,7], and determining the quality of both Q&A [8]. The post quality system has various applications like finding experts [9,10], information seeking [11], education [12], knowledge sharing [2,13], and question routing [14].

Accessing the quality of content posted on online communities is an active research area. In the existing research, user post quality can be determined by using features-based classification [8,11,15] and non-features-

*** Corresponding author: Zeeshan Anwar**, Department of Computer Software Engineering, National University of Sciences and Technology, H-12, Islamabad, Pakistan, e-mail: zeeshan.phdcse@students.mcs.edu.pk
Hammad Afzal, Naima Iltaf, Ayesha Maqbool: Department of Computer Software Engineering, National University of Sciences and Technology, H-12, Islamabad, Pakistan
Ali Ahsan: Centre for Healthy Sustainable Development, Torrens University, Adelaide SA 5000, Australia, e-mail: al_ahsan1@yahoo.com
ORCID: Zeeshan Anwar 0000-0002-8029-0604; Ali Ahsan 0000-0002-2023-0511

based classification [4,16]. Both of these techniques have their merits and demerits. However, feature-based techniques are most widely used for this purpose. For example, both Tóth et al. [17] and Baltadzhieva and Chrupallla [18] found that linguistic and semantic features, such as the presence of code snippets and the use of certain terms, can influence question quality. Mi et al. [19] found that the number of tags and code snippets is the most discriminative feature for identifying high-quality questions. Arora et al. [20] proposed a method for classifying questions based on retrieving similar questions previously asked in the same forum. These findings suggest that question quality on Stack Overflow can be assessed using various features and that automated methods can be used to improve the effectiveness of question quality moderation.

Furthermore, most of the researchers use binary classification to classify users' posts into good and bad categories [11,15]. The accuracy of these techniques is 74–76% [5,17]. But while analysing the StackOverflow posts, it is observed that the post should be categorized into four categories. Very good post has a high score and also accepted answer. A good question has a high score but no accepted answer. Similarly, a bad question has low score, and a very bad question has a low score and is also closed by the moderator. It is also observed that existing multi-classification techniques for user post quality assessment has a low accuracy. For example, the work of Kopp et al. has 41.6% accuracy [21]. Researchers also used different categories of features, including textual, format, community, and readability features [22]. Using the community features like reputation of user and score of question is biased towards the high reputation users [5]. The limitations of existing techniques are given in Table 1.

Table 1: Limitations of existing techniques proposed in the literature

Paper	Method	Classes	Limitations
[5]	LR	2	Accuracy of the proposed technique is low as authors used three features to train binary classifier.
[8]	Naive Bayes	2	Readability and structural features are used. The accuracy of technique is low.
[11]	NN	4	Author's popularity features are used that are biased towards high reputed authors, and only Precision of technique is reported.
[15]	GA	2	Author's popularity features are used that are biased towards high reputed authors, and only precision of technique is reported.
[17]	GRU	2	Structural features are used and accuracy of technique is low.
[23]	Co-predictions	2	Author's popularity features are used that are biased towards high reputed authors, and only improvement in error is reported.
[24]	LR	2	Question and answer features are used, and only accuracy of technique is reported.
[25]	RF, KNN, DT, XGBoost	2	This work predicts quality of edits only and uses author's features.
[26]	MLP, KNN, SVM	4	Format, textual, community and readability are used but accuracy is very low.
[27]	LOG-REG, SVM, NN	4	Accuracy of the classification is low and 10 features are used to train the classifier.
[28]	DT, RF, ANN, KNN, GNB	2	Main purpose of authors is to compare results of subjective and objective assessment of questions quality.

StackOverflow has developed a subjective assessment-based mechanism to determine the quality of Q&A. In subjective assessment score of posts rated by users is used to measure the quality of post. However, the objective mechanism that is based on different features that are calculated from post is better than the subjective mechanism as it is not biased [28]. While there have been previous attempts to evaluate the quality of user posts, these have various limitations as identified earlier. To overcome these limitations, a novel hybrid model for assessing the quality of StackOverflow posts is proposed. The proposed model combines the strengths of both convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. The proposed model is trained on a set of 20 different features that are extracted from the data. The feature extraction process is a critical step in accurately evaluating the quality of StackOverflow posts. Therefore, we selected different features and group these features into five different sets. Extensive experiments were performed to select the best features set. Finally, the best feature set is used to train the proposed classifier. The proposed hybrid model classifies the StackOverflow questions into four categories. For this task, already published and publicly available StackOverflow dataset [26] is used. Accuracy, precision, recall, and *F*1-score are used to evaluate models. In addition to this, the results of the proposed CNN + LSTM model are also compared with the existing state of the art and validated with various machine learning and deep learning models implemented and trained by authors. After the analysis of results it is observed that the proposed CNN + LSTM model gives the best accuracy on unseen test dataset that is considerably much higher than the baseline techniques.

The major contributions are enlisted as follows:

- Quality assessment of StackOverflow questions is performed using a novel hybrid deep learning architecture that combines the benefits of CNN and LSTM in one model. The proposed model and dataset is shared online for future research.
- Extensive experiments were performed, and a set of 20 features is proposed to access the quality of user posts.
- The performance of the proposed model has shown an improvement of 21–23% in accuracy as compared to the existing state of the art.

The remaining sections of this article are organized as follows. A review of related work is made in Section 2. Research methodology is explained in Section 3. Experimental results and analysis is given in Section 4. Validation of the proposed model is performed in Section 5. The proposed method is compared with state of the art tools in Section 6, and Section 7 concludes this article.

2 Related works

Extensive survey of literature is performed to conduct this research. In this section, articles that are most relevant to this research are reported. The authors highlighted and presented the tools and techniques, features, results, and limitations of existing research in this section.

Ponzanelli et al. [15] classified SO questions as good and bad quality questions. Readability metrics, author's popularity, and simple textual features of Stack Overflow are used. Classification is performed using decision trees and genetic algorithms. It is observed that decision trees are good to classify only on the author's popularity. Stanford NLP Parser is used to remove code snippets from questions. JGAP is used to implement genetic algorithm (GA). Important features are identified that determine the quality of questions. A precision of 80–97 is achieved using GA. It is concluded that the identification of bad-quality questions can be automated partially.

In another article, Ponzanelli et al. [11] proposed an approach to remove good-quality questions from the review queue, thus decreasing the workload of the moderators. Votes alone do not tell about the quality of the answer. Therefore, Omondiagbe et al. [29] used Random forest and neural network classifiers with various settings of features to determine the best feature set that can predict the accepted answer. It is observed that the length of code in the answer, the reputation of the user, the similarity of text between Q&A, and, the time lag between question & answer can best predict the accepted and unaccepted answers.

Ellmann and Schnecke [8] studied different measures that will increase the quality of question and answer. StackOverflow data dump of 2014 is used in this research. Error, discrepancy, and how-to question types are extracted from the dataset for further research. Naive Bayes with six features, i.e., Flesh-Reading-Ease score, Code-to-Text ratio, word count, image exists, listing exists, quote exists, and the number of tags are used to predict when a question will be answered or voted.

In another article [23], a co-prediction method based on regression and classification is proposed to predict the quality of questions and answers. The correlation between the quality of the question and answer is predicted. Features of question, answer, and author are used for this purpose. The model is trained on one million questions and 1.5 million answers. For the evaluation of model correlation, effectiveness, efficiency, root mean square error, and prediction error is used. This method improves 13.13% prediction error.

Rychwalska et al. [30] researched the association between communication and article quality at Wikipedia. The projects with denser communication between the editors are of high quality. For this purpose, the authors collected private communication between editors.

CQA sites evolved and contain a huge number of questions and answers. Reading all answers about a particular topic is not feasible and requires a huge effort. To reduce the number of answers, a technique based on grey wolf optimizer is proposed. The Biterm topic model is used to model the answer. This technique performs well to extract core answers [31].

There are three possibilities to create quality prediction systems, i.e. by learning features by using handcrafted features, by using deep learning to learn features, and by using a mixture of handcrafted and deep learning. The author uses CNN to learn the features and uses handcrafter features and CNN-generated features to learn deep neural network. This approach is called a deep fusion network. The authors use a total of 601 features. This technique is evaluated on SemEval-15 and SemEval-16, which consist of three and two classes, respectively. Accuracy of 75.24 and 76.67 is achieved on SemEval-15 and SemEval-16 datasets, respectively [16].

In a recent article [4], BERT is used to predict the quality of question and answer. MSDN dataset is used in the experiments, and many experiments using BERT are conducted. BERT as features extractor achieved an accuracy of 0.589, the BERT fine-tuning model achieved an accuracy of 0.696, and BERT pre-training model an achieved an accuracy of 0.774 on the test set.

Xiang et al. [32] developed two deep learning-based models for automatically tagging the answers in the SemEval-2015 dataset. The first model is based on CNN + LSTM and conditional random field. This model archives 58.96% *F*1 score. The second model is simple and is based on CNN and Bi-LSTM. The second model achieves 58.29% *F*1 score.

In another research [21], a dataset of 4,575 questions was manually labelled to judge the question quality posted on the intelligent learning system. The same dataset was processed by several NLP tools, and various NLP indices were calculated. These indices were used by the machine learning algorithm to predict the question quality. The results of the prediction were compared with the manually labelled dataset. Accuracy of 41.6% was achieved by using four classes and accuracy improved to 61.6% by predicting two classes of questions.

About 30% of the questions on StackOverflow are unanswered. To reduce this number and help the questioner to get answers to their question, Hsieh explored the literature, unanswered questions, and successful questions and collected features that make questions successful and unsuccessful. Three features named as code snippet presence, median length, and presence of attempt signifying words were selected by the author to train binary logistic regression classifier. A total of 75.59% accuracy, 76.92% precision, and 17.24% recall were attained. The author developed a chrome plugin using this approach. This plugin proposes answerability of the question before posting it to StackOverflow [5].

In a recent article, Roy and Singh [33] started by identifying several features that are indicative of a question being closed, such as the length of the question, the presence of certain words or phrases, and the number of tags associated with the question. The authors performed experiment with CNN and LSTM and then proposed a method that uses a CNN to automatically learn these features from the text of the question. The authors conducted experiments using a dataset from the Stack Exchange CQA site to evaluate their method. The results showed that their method outperformed existing methods for predicting closed questions, achieving an accuracy of over 80%. This method can help the users to improve the quality of their questions before posting to CQA sites.

Roy et al. [34] in another article identified several factors that are indicative of a question being closed, such as the length of the question, the number of words in all caps, and the presence of certain keywords. They then proposed a method that uses a Random Forest classifier to automatically learn these factors from the text of the question. The authors evaluate the performance of their method using a dataset from the Stack Exchange site and show that their method achieves an accuracy of over 80% in predicting question closability. They also conduct experiments to analyse the impact of different factors on question closability and find that factors such as the number of tags associated with the question and the length of the title have a significant impact on question closability.

In another article [17], researchers used linguistic and semantic features to train deep learning classifier to predict StackOverflow questions into good and bad quality. Title, body, and tags of questions are used. Embedding for Gated Rectifier Unit (GRU) was created by using Word2Vector and Doc2Vector. The authors performed four experiments by inputting different numbers of questions for training and gained a maximum of 74% accuracy.

A system based on an SVM classifier is proposed to find good and bad quality posts on Nabble.com. This approach uses surface, lexical, syntactic, and forum-specific features to determine the quality. This technique achieves 89% accuracy in categorizing posts into two classes. The authors experiment with different feature sets and found that without using forum-specific features, 82% accuracy can be achieved [22].

Ruseti et al. [35], used various variants of recurrent neural network (RNN) with different embedding to predict the quality of questions asked by students in an intelligent tutoring system. The accuracy of LSTM was low as compared to GRU and BiGRU. The best results in terms of accuracy, i.e., 81.22%, were obtained by using FastText embedding with the BiGRU network.

Selleras in his thesis [24] developed a model for identifying high quality and low answer in StackOverflow. The model consists of partial least square (PLS), natural language processing (NLP), and binomial logistic regression (BLR). Data for this research were extracted using stack exchange data explorer by using queries. Six features of the question and five features of the answer were selected. The regression equation was generated and the three Q&A were selected to test the model. The model predicted the optimal answer with 80% accuracy.

Recently, Mondal et al. [25] identified various reasons why edit on posts are rejected. To automate the edit quality assessment process, the authors trained four machine learning classifiers and developed a plugin. The accuracy of the proposed technique is 68%.

In another article, Mondal et al. [28] compared the results of subjective and objective assessments of post-quality. Subjective assessment is the default method adopted by StackOverflow in which users vote for the posts, and posts with a high score are considered as good quality. The authors identified various features and implemented five machine-learning algorithms to determine the objective quality of posts. The accuracy of the machine learning algorithm ranges from 68 to 86%.

The comparison of existing research in the form of their advantages and disadvantages is given in Table 2. Most of the existing researches [5,8,15,17,23,24] is based on binary classification. Classifying the user post quality into two classes increases the accuracy of the classifier, but the binary classifier may struggle to

Table 2: Advantages and disadvantages of existing techniques for posts quality assessment

Paper	Advantages	Disadvantages
[5]	Authors explored the literature and identified the features that make the question successful. Binary logistic regression is used to train the classifier	Developed plugin is not available online. Only three features are used, and accuracy is low.
[8]	Different measures are identified after review that increases the quality of questions and used Naive Bayes for binary classification	Only readability and structural features are identified by authors and the accuracy of the technique is low.
[11]	Features, i.e., length of code in answer, the reputation of the user, similarity of text between Q&A, and the time lag between question & answer, are used to train NN and predict the quality of the posts into four classes.	Author's features are biased towards high reputed authors and the code of this technique is also not available online.
[15]	GA is used to classify questions into good and bad categories. Readability, author's popularity and textual features are used and precision of up to 90% is achieved.	Author's popularity features are biased towards highly reputed authors and GA only works better with popularity features.
[17]	Linguistic and semantic features to train GRU and classify post in good and bad quality.	Structural features are used, and accuracy of the technique is low. The code of this technique is also not shared.
[23]	Co-predictions based method is used to find the relationship between the quality of question and answer. Different features of question, answer, and author are used.	Author's popularity features are biased towards highly reputed authors, and only prediction error is reported.
[24]	Logistic regression is trained to classify the Q&A into good and bad quality. The accuracy of this technique is 80%.	Question and answer features are used, and only the accuracy of the technique is reported. The code of this technique is not shared.
[25]	Identified 17 features and trained machine learning classifier to predict the quality of post edits.	Author features are biased, and the accuracy of proposed classifiers is low.
[28]	Authors work shows that objective assessment of quality is better than subjective assessment	This work is based on binary classification using various traditional machine learning classifiers.

compute the quality of questions with accepted answers and closed questions. Therefore, a classifier that can predict the quality of users' posts into four classes is more suitable. The accuracy of the existing classifiers [11,15] is low. To overcome these limitations, CNN, and LSTM-based hybrid model is developed for the prediction of user quality into four categories.

3 Proposed research methodology

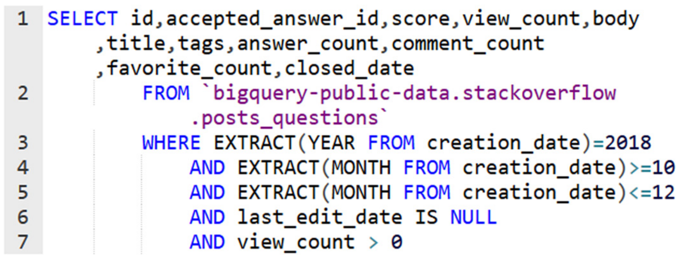
The proposed research methodology is divided into five major steps: data collection from StackOverflow, data pre-processing & cleaning, features extraction, developing & training the proposed model, and making predictions from trained models on new test data.

3.1 Data collection

StackOverflow data can also be accessed by using Google BigQuery [36]. By using BigQuery, one does not need to download all the data dump and the required data can be extracted easily and quickly. For this research, the data were extracted from the StackOverflow posts table. The extracted fields and their description are given in Table 3, and the query used to extract data is given in Figure 1.

Table 3: Description of data fields extracted using BigQuery and used to predict quality

Field Name	Description
Id	Unique identifier of posts/questions
Accepted Answer Id	Unique identifier of accepted answer
Score	Score of question given by community member.
View count	Total views of question
Body	Body/text of the question
Title	Title of the question
Tags	Tags of question
Answer count	Total number of answer for specific question
Comment count	Total number of comments posted for specific answer
Favorite count	Number the question was favourite
Close date	Date on which question was closed



```
SELECT id,accepted_answer_id,score,view_count,body
    ,title,tags,answer_count,comment_count
    ,favorite_count,closed_date
        FROM `bigquery-public-data.stackoverflow
            .posts_questions`
        WHERE EXTRACT(YEAR FROM creation_date)=2018
            AND EXTRACT(MONTH FROM creation_date)>=10
            AND EXTRACT(MONTH FROM creation_date)<=12
            AND last_edit_date IS NULL
            AND view_count > 0
```

Figure 1: Google BigQuery used to extract data from Stackoverflow.

To make this research comparable and replicate the results of the existing state-of-the-art methods, the same query is used to extract data as used in [26,27,37]. A total of 200,000 questions were extracted. 70/30 rule is used to randomly split the dataset into training and testing sets. 10% of training data is reserved for validation.

3.2 Data pre-processing

In the pre-processing phase, firstly, the results of BigQuery are converted into CSV format from XML format. The data are typecast so that it can be easily processed and features can be extracted. Data were labelled into four categories, i.e., very good, good, bad, and very bad questions. This categorization is made based on the score of questions. The score of questions is also used by [26,37] to categorize the data. A description of each category and criteria to categorized questions is given in Table 4.

Table 4: Quality classification criteria

Total Questions	Criteria	Class
115,302	Score is >0 and has accepted answer	Very good
70,411	Score is >0	Good
1,781	Score is <0	Bad
12,506	Score is <0 and is closed or deleted	Very bad

3.3 Feature extraction

To determine the optimal set of features for StackOverflow post quality assessment. Multi layer perceptron term frequency (MLP-TF) model was implemented, and experiments were conducted with different sets of features, i.e. 7, 10, 12, 17, and 20 list as given in Table 5. A set of 20 features from the dataset is extracted using Python script after extensive experiments with different sets because this set gives the best results. The selected features are divided into four categories. The detail of each feature and their category is given below:

Table 5: Experiments with different combinations of features

Number of features	List of features	Accuracy
7	URL Count, body text length, code percentage, FLESH, Coleman, automated readability index, gunning	72.34
10	Title length, question length, body text length, average word length, average sentence length, uppercase percentage, lowercase percentage, code percentage, Coleman, Dale Chall	73.12
12	Number of tags, title length, URL count, paragraph count, list count, question length, body text length, average word length, average sentence length, uppercase percentage, lowercase percentage, code percentage	74.86
17	Number of tags, title length, URL count, paragraph count, list count, question length, body text length, average word length, average sentence length, uppercase percentage, lowercase percentage, code percentage, flesch, smog, automated readability index, Dale Chall, difficult words	76.89
20	Number of tags, title length, URL count, paragraph count, list count, question length, body text length, average word length, average sentence length, uppercase percentage, lowercase percentage, code percentage, Flesch, Smog, Kincad, Coleman, Automated readability index, Dale Chall, gunning, difficult words	78.00

3.3.1 Format features

Three format or structure-related features (Paragraph count, List count, and URL count) from the body of the questions were extracted. These features were extracted by using the Python library BeautifulSoap [38]. Paragraph count is the total number of paragraphs in the question body, List count is the number of lists in the body, and URL count is the number of URLs refereed in the question body.

3.3.2 Textual features

Seven textual features are extracted from the question title and body. Details of each of these features and the calculation process are given in Table 6.

Table 6: Textual features

Feature	Description
Title length	Total number of characters in title
Question length	Total number of characters in question body including code
Body text length	Total number of characters in question body
Average word length	$\mathrm{WL}_{\mathrm{Avg}} = \frac{\mathrm{TC}}{\mathrm{TW}} \times 100$, TC is total characters and TW is total words
Average sentence length	$\mathrm{SL}_{\mathrm{Avg}} = \frac{\mathrm{TW}}{\mathrm{TS}} \times 100$, TW is total words and TS is total sentences
Uppercase percentage	$U_{\mathrm{CASE}} = \frac{\mathrm{UC_L}}{\mathrm{T_L}} \times 100$, $\mathrm{UC_L}$ is uppercase letters and $\mathrm{T_L}$ is total letters
Lower case percentage	$\mathrm{L}_{\mathrm{CASE}} = \frac{\mathrm{LC_L}}{\mathrm{T_L}} \times 100$, LC_L is lowercase letters and T_L is total letters

3.3.3 Community features

Community features are specific to the community website. Every community question-answering site has different community features. To extend this research to other community sites, only two community features are used because these features are available in almost all community question-answering sites. The number of tags is the total number of tags that are associated with the question and the code percentage is the questions' text-to-code ratio that can be calculated by equation (1). *CS* is a number of code sentences, and TS is a number of text sentences.

$$\mathrm{Code}_{\mathrm{Per}} = \frac{\mathrm{CS}}{\mathrm{TS}} \times 100. \tag{1}$$

3.3.4 Readability features

Readability features measure the readability of a text. Eight readability features to measure the readability of the StackOverflow questions were calculated by using TextStat Python Library [39]. The detail of each feature is given below:

The flesch reading ease (FRES) counts the words, syllables, and sentences in the given text and calculates the average number of words per sentence and the average number of syllables per word. A higher FRES score means that text is easier to understand because shorter words and sentences are easier to read. FRES can be calculated by using equation (2):

$$\mathrm{FRES} = 206.835 - 1.015\left(\frac{\text{Total words}}{\text{Total sentences}}\right) - 84.6\left(\frac{\text{Total syllables}}{\text{Total words}}\right). \tag{2}$$

SMOG index or SMOG grade estimates the years of education needed to understand the text. It can be calculated by equation (3).

$$\mathrm{SMOG} = 1.0430\sqrt{\text{No of poly syllables} \times \frac{30}{\text{No of sentences}}} + 3.1291. \tag{3}$$

Flesch-Kincaid grade level also determines the years of education required to understand the text. It is for the U.S. grade level. It can be calculated from equation (4).

$$\text{KINCAID} = 0.39\left(\frac{\text{Total words}}{\text{Total sentences}}\right) - 15.59\left(\frac{\text{Total syllables}}{\text{Total words}}\right). \tag{4}$$

Coleman-Liau index is a readability test that measures the understandability of a text. It can be determined by equation (5).

$$\text{CLI} = 0.0588L - 0.296S - 15.8, \tag{5}$$

where L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

Automated readability index is also a readability test designed to measure the understandability of a text. Its formula is given in equation (6).

$$\text{ARI} = 4.71\left(\frac{\text{Characters}}{\text{Words}}\right) + 0.5\left(\frac{\text{Words}}{\text{Sentences}}\right) - 21.43. \tag{6}$$

Dale-Chall readability score is used to judge the comprehension difficulty faced by readers. Its formula is given in equation (7).

$$\text{DALE} = 0.1579\left(\frac{\text{Diff words}}{\text{Words}} \times 100\right) + 0.0496\left(\frac{\text{Words}}{\text{Sentences}}\right). \tag{7}$$

Fog scale determines the years of education required to understand the text. It can be calculated by equation (8).

$$\text{FOG} = 0.4\left[\left(\frac{\text{Words}}{\text{Sentences}}\right) + 100\left(\frac{\text{Complex words}}{\text{Words}}\right)\right]. \tag{8}$$

Difficult words are also determined by using TextStat Libarty. Difficult words are those having syllables >2.

3.4 Proposed CNN, and LSTM-based classification model for StackOverflow post quality assessment

To further improve the accuracy of StackOverflow questions quality assessment, a hybrid model of CNN and LSTM is proposed and implemented. The intuition behind the selection of this hybrid model is that after various experiments with machine learning and deep learning model results of this model are quite high. Furthermore, this model achieved state-of-the-art accuracy on different tasks and StackOverflow dataset [40,41]. By using the hybrid of CNN and LSTM advantages of both models can be gained. CNN model helps us to extract the features and LSTM helps to capture the long-term dependencies in the data [42]. The architecture of the proposed model is given in Figure 2. The input to the model is set to 20× selected features. The model consists of two Conv1D layers with 64 filters and 3 kernel size. After Conv1D layers, dropout of 0.5 is added to avoid over-fitting. The Maxpooling layer after the dropout is also added to the down-sample input representation. Maxpooling layer is followed by the LSTM layer with 100 units. The next layer is a dense layer with 100 units. All of these layers are activated by the relu activation function. The last layer is dense with 4 units which is the output layer. The output layer is activated by the softmax activation function. The model is compiled with Adam optimizer and the loss function is categorical cross-entropy.

3.5 Classification

The CNN + LSTM based model for classification of StackOverflow posts is tested on a test dataset that is 30% of the entire dataset. The test dataset is after pre-processing, and features extraction is given to the model as input and the model predicts the class of question.

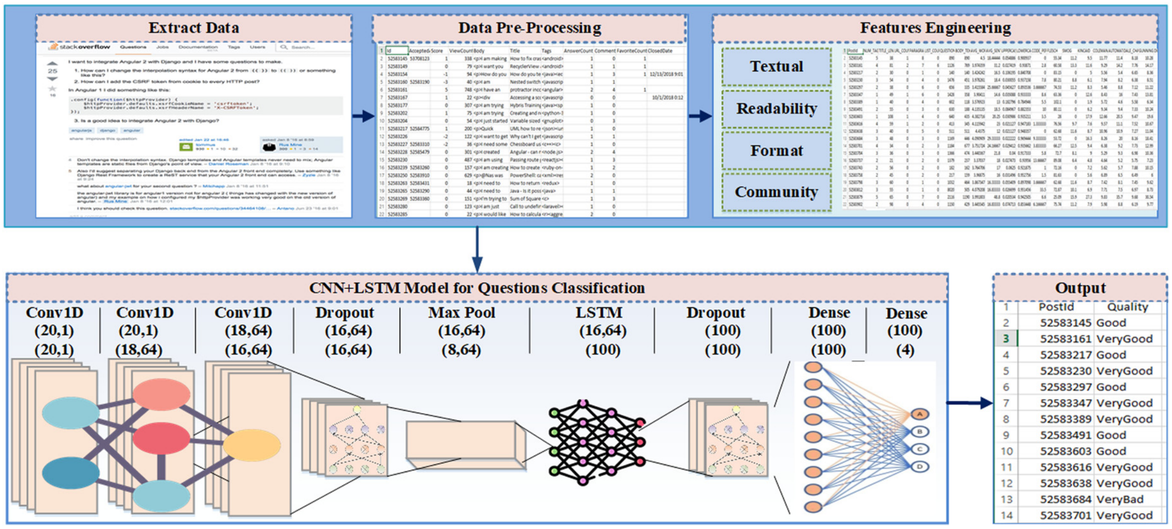


Figure 2: Proposed CNN + LSTM model-based architecture for StackOverflow question quality assessment.

4 Results and discussion

The proposed model of CNN and LSTM is implemented in Google Colab [43] environment by using Keras with Tensorflow backend. Several experiments with different numbers of CNN and LSTM layers were conducted, and the accuracy of each experiment was determined. The best results were gained by the model as given in Figure 2. This model is used to report the results. As stated earlier, the optimal set of features was identified, and the final model using the 20 features is trained on processed training data and tested on processed test data. The dataset is explained in Section 3.1, and data processing steps are explained in Section 3.2. The output of the model is the quality of the question and four metrics, i.e. accuracy, precision, recall, and *F*1-score that are used to analyse the results of the prediction model. The equations to calculate these metrics are given as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{11}$$

$$F1\ \text{score} = \frac{2{*}\text{Precision}{*}\text{Recall}}{\text{Precision} + \text{Recall}}. \tag{12}$$

In the aforementioned equations, TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

The results of the proposed hybrid model that is given in Table 7. The accuracy of the model is 80.45%, which is considerably high and indicates that the model makes 80.45% correct predictions, which is a signification portion of the dataset. The precision of the model is 57.65%, which means the model is making only 57.65% true positive predictions from the overall positive predictions. Again the recall of the proposed model is 57.65%, which means the model is correctly predicting 57.65% of actual positive classes. The *F*1-score of the model indicates the balance between precision and recall, but in this case, the model is struggling to find the right trade-off between precision and recall. The accuracy, precision, recall, and *F*1-score of the model are given in Figure 3.

The accuracy of the model is acceptable and higher than all of the models implemented and proposed in the literature. However, precision, recall, and *F*1-score indicate that there is room for significant improvement

Table 7: Algorithms results on test dataset

Algorithm	**Accuracy**	**Precision**	**Recall**	***F*1-score**
MLP	56.53	30.19	27.49	26.39
KNN	56.25	24.89	25.88	23.16
SVM	57.65	14.41	25.0	18.28
MLP-TF	78.00	57.65	57.65	57.65
LSTM	78.56	46.43	92.82	61.90
GRU	78.56	46.42	92.84	61.89
CNN	78.82	57.65	57.65	57.65
CNN+GRU	80.32	57.65	57.65	57.65
CNN + LSTM	**80.45**	**57.65**	**57.65**	**57.65**

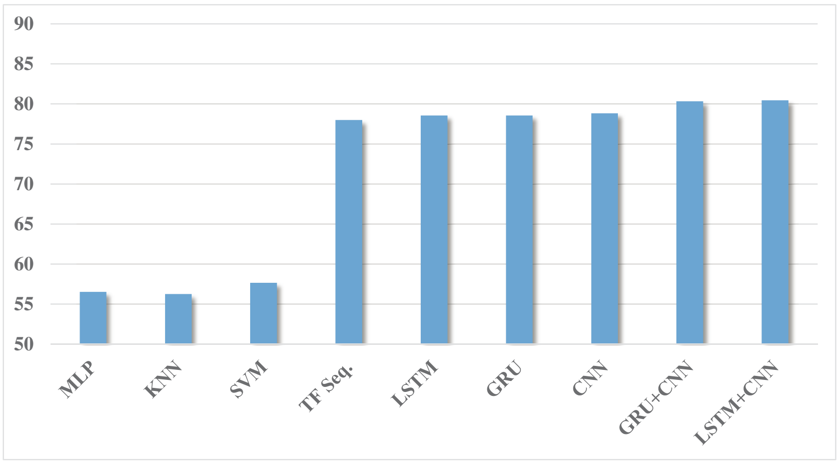


Figure 3: Comparison between accuracy of different models for post quality assessment.

in the model. The possible improvement in the model can be achieved by optimizing the model hyperparameters that are already tested by using different numbers of layers, dropout, and max pooling. The second option is to select different feature sets that are already experimented with by different combinations of features. The third possibility is choosing a different model that is also experimented with by four different deep learning models whose detail is given in the model validation section. Finally, the balanced dataset can be used for training the model which may improve the performance of the model.

5 Model validation

Several machine learning and deep learning classifiers are trained to predict the quality of questions and validate the proposed model. The input of each model is 20 features, and the output is a class of questions defined as Very Good, Good, Bad, and Very Bad. A total of 70% data is used for training, 30% data for testing, and 10% data for validation of the model. A detail of each model that is trained with hyper-parameters, and results are given in Appendix A.

The results of all the nine models that are used in experiments are given in Table 7. From the results, one can observe that SVM, KNN, and simple MLP,models are not suitable for judging the quality of StackOverflow questions because of low accuracy. The accuracy of all of these three models is low as compared to deep learning models. But when the number of layers is increased in MLP, there is a significant improvement in the accuracy that can be observed by analysing the Figure 3. When complex models are trained the accuracy is also improving. For example, the accuracy of the MLP-TF model is 78%, whereas, the accuracy of the CNN + LSTM model is 80.45%. The best accuracy of 80.45% is achieved by using the CNN + LSTM model.

From the results, one can observe that SVM, KNN, and simple MLP models are not suitable for judging the quality of StackOverflow questions because of low accuracy. The accuracy of all of these three models is low as compared to deep learning models. But when the number of layers is increased in MLP, there is a significant improvement in the accuracy that can be observed by analysing the Figure 3. When complex models are trained, the accuracy is also improving. For example, the accuracy of the MLP-TF model is 78%, whereas the accuracy of the CNN + LSTM model is 80.45%. The best accuracy of 80.45% is achieved by using the CNN + LSTM model.

After analysing the precision, recall, and *F*1-score of the models, one can observe that these metrics of the four deep learning models are the same. There is a sudden increase in precision, recall, and *F*1-score when complex deep learning models are trained, but after changing the models and using hybrid model, values of these metrics do not change. The precision, recall, and *F*1-score of MLP-TF, CNN, GRU+CNN, and LSTM+CNN models is 57.65%. The recall and *F*1-score of RNN models are high as compared to other deep learning models. But accuracy and precision of RNN models are lower than the other deep learning models. Therefore, one can say that the best model for the assessment of StackOverflow questions quality is the LSTM+CNN model as the accuracy of this model is high, and precision, recall, and *F*1-score of this model is also consistent with other deep learning models, i.e. TF Seq., CNN, and GRU+CNN.

6 Comparison and evaluation

The proposed hybrid model of CNN + LSTM for the quality assessment of StackOverflow questions is compared with various existing models reported in the literature [5,8, 11,15,17, 23–26]. Most of the existing techniques for quality assessment are developed by using binary classifiers. The maximum precision of 80–90% is reported by Ponzanelli et al. [15] with binary classification. The results of the comparison are given in the Table 8. In most of the binary classification techniques, machine learning classifiers are used. Only one researcher used a deep learning model, i.e. gated recurrent unit (GRU), and obtained 74% accuracy.

Table 8: Comparison with state of the art

Algorithm	Classes	Features	Results
LR [5]	Binary	Code snippet, median length and presence of attempt signifying words	75.59% accuracy
Naive Bayes [8]	Binary	Flesh-Reading-Ease, Code-to-Text ratio, word count, image exists, listing exists, quote exists and number of tags	62.51% accuracy
GA [15]	Binary	Readability, author's popularity, and textual features	80-97% precision
GRU [17]	Binary	Title, body and tags	74% accuracy
CoPs (Co-Predictions) [23]	Binary	question, answer and author	13.13% improvement in error
DT, RF, ANN, KNN, GNB [28]	Binary	Title Quality, Text Readability, Code Readability, Text Code Ratio, Text Code, Co-relation, Code Re-usability, Code Understand-ability, Topic Entropy, Metric Entropy, Sentiment Polarity	68-86% accuracy
PLS, NLP and BLR [24]	Binary	Six features of question and five features of answer	80% accuracy
NN [11]	Four	Code Length, user reputation, Q&A similarity and time lag between Q&A	60.67% precision
MLP, KNN, SVM [26]	Four	Format, Textual, Community and Readability	56.53% accuracy
LOG-REG, SVM, NN [27]	Four	ARI, FRE, CLI, Gunning Index, Body Length, Presence of URL, Code to Text Ratio and Title Body Similarity	58.75% accuracy
Proposed	Four	20 features	80.45% accuracy

In the literature, only three techniques [11,26,27] uses multi-class classifier (four classes) to classify the question's quality. These techniques are similar to this work, but out of these two techniques, Ponzanelli et al. [11] used the features of answers to judge the quality of the question. This is based on the assumption that good questions are always answered by community members. This technique is good only for classifying the old questions because answers to new questions are not always available to extract features. During analysis of the literature, it is observed that a technique that can classify the questions at the time of posting is more valuable and will help the moderators to make decisions about the question. Therefore, the focus of this research is to classify the questions without using the features of the answer. The rest of the two techniques, i.e. the studies by Vakil and Mahajan [26,27], are more relevant to this work and use the same dataset and features to assess the quality of StackOverflow questions.

In the first technique, Vakil [26] used three classifiers, i.e. MLP, KNN, and SVM to assess the quality of StackOverflow questions. The best results were achieved by using MLP with 56.53% accuracy, 30.19% precision, 27.49% recall, and 26.39% *F*1-score. As compared to these results, our proposed model is 23.92% accurate, and the *F*1-score of the proposed model is also 31.26% better than the results of Vakil.

In the second technique [27], NN, SVM, and logistic regression are used for the assessment of StackOverflow questions quality. This technique also uses the subset of the dataset that is used in this research. The authors use seven features, i.e. ARI, FRE, CLI, gunning index, body length, presence of URL, code to text ratio and title body similarity in their research. Six of these features are also used in this research. The additional features that are used by authors are title and body similarity. The authors used cross-validation to implement all three algorithms and reported the accuracy of their algorithms. The maximum average accuracy of 58.75% is achieved by using SVM and NN classifiers on training data. Results on the test dataset are not reported by the authors. The accuracy of the author's model is 21.7% lower than accuracy of our proposed CNN + LSTM method, which is computed on the test dataset.

After the comparison, it can be observed that existing models are based on traditional machine learning models and the accuracy of these models is also low. This research is novel in the sense that the existing problem is solved by developing a novel hybrid model of CNN and LSTM. The results of the proposed model are also much higher than the existing techniques.

7 Conclusion and future work

In this research, firstly, an extensive literature review of various articles related to the quality assessment of users' posts is performed to get insight into the latest trends and find the research gap. From the existing research, it is observed that features-based classification techniques are most popular for the quality assessment of user posts, but an optimal set of features for this purpose is not known. Therefore, the first goal of this research is to select the optimal set of features. Different features that are used in the existing research studies were listed and divided into various categories. After this extensive experiments were performed for the selection of an optimal set of features. After experiments, a set of 20 features were selected for further implementation. In the literature, most of the techniques implemented for content quality assessment are binary classification techniques. While deeply analysing the StackOverflow posts, it is observed that StackOverflow posts quality is classified into four classes. Therefore, existing binary classifiers do not correctly reflect the actual quality of StackOverflow posts. Above in view, a hybrid multi-classifier is implemented for the classification of SO posts. In this article, various experiments with different features and machine learning algorithms are performed to classify StackOverflow questions based on the quality of the questions. After various experiments, 20 features are selected and a hybrid deep learning model that is a combination of CNN and LSTM is developed to classify the questions. The model's performance is compared with the various other deep learning and machine learning models. The accuracy of the model is higher than all of the other models and state-of-the-art methods proposed in the literature. The proposed model is 23.92% accurate as compared with [26] and 21.7% accurate as compared to [37].

In the future, plan to further improve the accuracy of the proposed model by using different features, classifiers, and data balance methods as there is huge potential to further improve the accuracy. This will improve the recall and *F*1-score. Testing the performance of the proposed model on other software engineering community question-answering datasets like Quora and MSDN questions is also included in our future plan.

Funding information: The author states no funding is involved.

Author contributions: In this research, all the authors contributed equally. Zeeshan Anwar analyzed the gap in the existing research and prepared the proposal. Hammad Afzal and Naima Iltaf reviewed and approved the research proposal. Zeeshan Anwar implemented the various algorithms and prepared the initial draft of article. Ali Ahsan and Ayesha Maqbool performed the various review iterations and helped to improve the quality of the article. Initial results were prepared by Zeeshan Anwar that were analyzed by Hammad Afzal and Ali Ahsan. Validation with the exiting algorithms was performed by the Naima Iltaf and Ayesha Maqbool. Zeeshan Anwar performed various revisions of article as per directions of Hammad Afzal. Ali Ahsan arranged the funds to pay APC.

Conflict of interest: The authors state no conflict of interest.

Ethical approval: The research not related to human use complies with all the relevant national regulations and institutional policies.

Informed consent: Informed consent has been obtained from all individuals included in this study.

Data availability statement: Data is already publically available. Its source is cited in the paper.

References

[1] Hanrahan BV, Convertino G, Nelson L. Modeling problem difficulty and expertise in stackoverflow. In: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion. ACM; 2012. p. 91–4.

[2] Calefato F, Lanubile F, Marasciulo MC, Novielli N. Mining successful answers in stack overflow. In: 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories. Florence: IEEE; 2015. p. 430–3.

[3] Le LT, Shah C, Choi E. Evaluating the quality of educational answers in community question-answering. In: 2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL). IEEE; 2016. p. 129–38.

[4] Sen B, Gopal N, Xue X. Support-BERT: predicting quality of question-answer pairs in MSDN using deep bidirectional transformer. 2020. arXiv: http://arXiv.org/abs/arXiv:200508294.

[5] Hsieh JW. Asking questions is easy, asking great questions is hard: constructing effective stack overflow questions. Oberlin, United States: Oberlin College; 2020.

[6] Roy PK, Ahmad Z, Singh JP, Alryalat MAA, Rana NP, Dwivedi YK. Finding and ranking high-quality answers in community question answering sites. Global J Flexible Syst Manag. 2018;19:53–68.

[7] Roy PK, Ahmad Z, Singh JP, Banerjee S. Feature extraction to filter out low-quality answers from social question answering sites. IETE J Res. 2022:1–12. doi: 10.1080/03772063.2022.2048715.

[8] Ellmann M, Schnecke M. Two perspectives on software documentation quality in stack overflow. In: Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering. NY, USA: ACM; 2018. p. 6–9.

[9] Fu J, Li Y, Zhang Q, Wu Q, Ma R, Huang X, et al. Recurrent memory reasoning network for expert finding in community question answering. In: Proceedings of the 13th International Conference on Web Search and Data Mining; 2020. p. 187–95.

[10] Yuan S, Zhang Y, Tang J, Hall W, Cabotà JB. Expert finding in community question answering: a review. Artif Intelligence Rev. 2020;53(2):843–74.

[11] Ponzanelli L, Mocci A, Bacchelli A, Lanza M, Fullerton D. Improving low quality stack overflow post detection. In: IEEE International Conference on Software Maintenance and Evolution. Piscataway, NJ, USA: IEEE Press; 2014. p. 541–4.

[12] Joorabchi A, English M, Mahdi AE. Text mining stackoverflow. J Enterprise Inform Manag. 2016 March;29(2):255–75.

[13] Vasilescu B, Serebrenik A, Devanbu P, Filkov V. How social Q&A sites are changing knowledge sharing in open source software communities. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing. ACM; 2014. p. 342–54.

[14] Chang S, Pal A. Routing questions for collaborative answering in community question answering. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM; 2013. p. 494–501.

[15] Ponzanelli L, Mocci A, Bacchelli A, Lanza M. Understanding and classifying the quality of technical forum questions. In: 4th International Conference on Quality Software. Piscataway, NJ, USA: IEEE Press; 2014. p. 343–52.

[16] Suggu SP, Goutham KN, Chinnakotla MK, Shrivastava M. Deep feature fusion network for answer quality prediction in community question answering. 2016. arXiv: http://arXiv.org/abs/arXiv:160607103.

[17] Tóth L, Nagy B, Janthó D, Vidács L, Gyimóthy T. Towards an accurate prediction of the question quality on stack overflow using a deep-learning-based NLP approach. In: 14th International Conference on Software Technologies, ICSOFT 2019. SciTePress; 2019. p. 631–9.

[18] Baltadzhieva A, Chrupallla G. Predicting the quality of questions on stackoverflow. In: Proceedings of the International Conference Recent Advances in Natural Language Processing; 2015. p. 32–40.

[19] Mi Q, Gao Y, Keung J, Xiao Y, Mensah S. Identifying textual features of high-quality questions: an empirical study on stack overflow. In: 2017 24th Asia-Pacific Software Engineering Conference (APSEC). IEEE; 2017. p. 636–41.

[20] Arora P, Ganguly D, Jones GJ. The good, the bad and their kins: Identifying questions with negative scores in stackoverflow. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015; 2015. p. 1232–9.

[21] Kopp KJ, Johnson AM, Crossley SA, McNamara DS. Assessing question quality using NLP. In: International Conference on Artificial Intelligence in Education. Springer; 2017. p. 523–7.

[22] Weimer M, Gurevych I, Mühlhäuser M. Automatically assessing the post quality in online discussions on software. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions; 2007. p. 125–8.

[23] Yao Y, Tong H, Xie T, Akoglu L, Xu F, Lu J. Want a Good Answer? Ask a Good Question First! 2013. ArXiv e-prints. http://arxiv.org/abs/1311.6876.

[24] Selleras RQ. Predictive model: using text mining for determining factors leading to high-scoring answers in stack overflow. New York City, United States: The George Washington University; 2020.

[25] Mondal S, Uddin G, Roy C. Automatic prediction of rejected edits in stack overflow. Empirical Softw Eng. 2023;28(1):9.

[26] Vakil Y. CQA-Question-Quality. Accessed: 2020-07-11. https://github.com/yashvakil/CQA-Question-Quality.

[27] Mahajan V. CQA stackoverflow predicting question quality and auto-tagging. Accessed: 2020-07-11. https://github.com/vatsal13/CQA-Stackoverflow-Predicting-Question-Quality-and-Auto-Tagging.

[28] Mondal S, Rahman MM, Roy CK. Do subjectivity and objectivity always agree? a case study with stack overflow questions. 2023. arXiv: http://arXiv.org/abs/arXiv:230403563.

[29] Omondiagbe OP, Licorish SA, MacDonell SG. Features that predict the acceptability of java and JavaScript answers on stack overflow. In: Proceedings of the Evaluation and Assessment on Software Engineering. New York, NY, USA: ACM; 2019. p. 101–10.

[30] Rychwalska A, Talaga S, Ziembowicz K. Quality in peer production systems-impact of assortativity of communication networks on group efficacy. In: Proceedings of the 53rd Hawaii International Conference on System Sciences; 2020.

[31] Li M, Chen L, Chen Y, Wang J. Extracting core answers using the grey wolf optimizer in community question answering. Appl Soft Comput. 2020;90:106125.

[32] Xiang Y, Zhou X, Chen Q, Zheng Z, Tang B, Wang X, et al. Incorporating label dependency for answer quality tagging in community question answering via cnn-lstm-crf. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers; 2016. p. 1231–41.

[33] Roy PK, Singh JP. Predicting closed questions on community question answering sites using convolutional neural network. Neural Comput Appl. 2020;32(14):10555–72.

[34] Roy PK, Singh JP, Banerjee S. Is this question going to be closed? Answering question closibility on Stack Exchange. J Inform Sci. 2022:01655515221118665. doi: 10.1177/01655515221118665.

[35] Ruseti S, Dascalu M, Johnson AM, Balyan R, Kopp KJ, McNamara DS, et al. Predicting question quality using recurrent neural networks. In: International Conference on Artificial Intelligence in Education. Springer; 2018. p. 491–502.

[36] Tigani J, Naidu S. Google big query analytics. Hoboken, New Jersey: John Wiley & Sons; 2014.

[37] Aggarwals S. MiningCQAData. Accessed: 2022-07-11. https://github.com/SurbhiAggarwal04/MiningCQAData.

[38] Richardson L. Beautiful soup documentation. April. 2007.

[39] Huning M, Bennett G. TextSTAT 2.7 Useras Guide. TextSTAT, created by Gena Bennett. 2007.

[40] Tabassum J, Maddela M, Xu W, Ritter A. Code and named entity recognition in stackoverflow. 2020. arXiv: http://arXiv.org/abs/arXiv:200501634.

[41] Batchuluun G, Yoon HS, Kang JK, Park KR. Gait-based human identification by combining shallow convolutional neural network-stacked long short-term memory and deep convolutional neural network. IEEE Access. 2018;6:63164–86.

[42] Rehman AU, Malik AK, Raza B, Ali W. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools Appl. 2019;78(18):26597–613.

[43] Google Colaboratory. Accessed: 2022-11-23. https://colab.research.google.com/notebooks/intro.ipynb.

[44] Joachims T. Learning to classify text using support vector machines. vol. 668. New York City, United States: Springer Science & Business Media; 2002.

[45] Hsu CW, Lin CJ. A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw. 2002;13(2):415–25.

[46] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Machine Learn Res. 2011;12:2825–30.
[47] Murtagh F. Multilayer perceptrons for classification and regression. Neurocomputing. 1991;2(5–6):183–97.
[48] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization. 2014. arXiv: http://arXiv.org/abs/arXiv:14092329.
[49] Ali Reshi J, Ali R. An efficient fake news detection system using contextualized embeddings and recurrent neural network. Int J Interactive Multimedia Artif Intell. 2023;1–13.
[50] Cavaliere D, Fenza G, Loia V, Nota F. Emotion-Aware Monitoring of Users Reaction With a Multi-Perspective Analysis of Long-and Short-Term Topics on Twitter. Int J Interactive Multimedia Artif Intell. 2023;1–10.
[51] Jianqiang Z, Xiaolin G, Xuejun Z. Deep convolution neural networks for twitter sentiment analysis. IEEE Access. 2018;6:23253–60.
[52] Ye Z, Saleem N, Ali H, et al. Efficient gated convolutional recurrent neural networks for real-time speech enhancement. Int J Interactive Multimedia Artif Intell. 2023;1–8.
[53] QDN. Qualcomm Developer Network, Deep Learning and Convolutional Neural Networks for Computer Vision. Accessed: 2023-03-01. https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/cnn-architectures/deep-learning-convolutional-neural-networks-computer-vision.

Appendix

A Model validation with machine learning and deep learning classifiers

In this research, various machine learning and deep learning classifiers are implemented for the validation of the proposed hybrid model for the classification of StackOverflow questions based on its quality. A quick reference to the classifiers used in this research is given below:

A.1 Support vector machines (SVM)

SVM [44], was created for binary classification and is a linear non-probabilistic classifier. SVM can also be extended for multi-class classification [45]. In SVM, the goal is to find the maximum-margin hyperplane (equation (A1)) for a training set (x_i, y_i) that divides the points with $y_i = 1$ and $y_i = -1$.

$$w.\ xb = 0. \tag{A1}$$

The equation to maximize the margin λ is given in equation (A2).

$$\max_{\omega,\gamma} \gamma, \quad \text{s.t. } \forall i, \gamma \leq y_i(w.\ x_i + b). \tag{A2}$$

In this article, SVM is used as a baseline method and implemented by using Python Scikit-learn library [46]. The implementation of SVM can be found in the study by Vakil [26]. SVM was trained with different values of hyper-parameters, i.e. Kernel, error term (C term), and gamma. The best results (accuracy = 57.65) were attained by using Kernel = rbf, $C = 1.0$, and Gamma = 0.0001. The results of SVM for question quality assessment are not satisfactory. The complete results of SVM are given in Table 7.

A.2 K-nearest neighbour (KNN)

KNN is a non-parametric algorithm that is used for regression and classification. The class in KNN is decided by the polarity vote of its members. The distance between neighbours can be determined by using various distance formulas. The formula for Manhattan distance is given in equation (A3).

$$d = \sum_{i=i}^{k} |x_i - y_i|. \tag{A3}$$

In the experiments, KNN is also used as a baseline method and implementation of KNN can be found in the study by Vakil [26]. KNN is implemented by using Scikit-learn [46] and trained with different numbers of neighbors, i.e. 31–43. The best accuracy of 56.25% was achieved by using 43 neighbours. The results of KNN are given in Table 7.

A.3 Multilayer perceptron (MLP)

Multilayer perceptron (MLP) [47] is the third baseline method in this research. MLP is a feed-forward neural network with at least three layers. In MLP, each neuron uses a linear activation function and learns by using a backpropagation learning algorithm. The output of MLP can be calculated using equation (A4).

MLP performs well in complex classiïňĄcation problems by learning non-linear models such as sentiment analysis.

$$y = \varphi\left(\sum_{i=1}^{n} w_i xi + b\right) = \varphi(W^T X + b), \tag{A4}$$

where w is a vector of weights, x is the input vector, b is bias, and φ is the activation function.

In this research, baseline implementation of MLP [26] is used, which is implemented by using Scikit-learn [46] with 100 hidden units, relu activation, and adam solver. The training cycle was repeated for 1,000 iterations, and loss improvement for 10 iterations is used to stop training. It is observed that after 259 iterations loss does not improve. The results of MLP as shown in Table 7 are better than SVM. The accuracy of MLP on the test dataset is 56.53%, precision is 30.19%, recall is 27.49%, and *F*1-score is 26.39%.

Multilayer perceptron using tensorflow and Keras is also implemented as this model suits tabular data where each column represents a feature. This model consists of four layers, and the input of each layer is fully connected with the next layer. The input layer of this model is the features layer, the second layer consists of 256 hidden units, the third layer consists of 128 hidden units, and the fourth layer is the output layer that consists of four units. Four hidden layers relu activation function is used, whereas for output layer, sigmoid activation function is used. The loss function of the model is categorical cross-entropy. The model is trained for five epochs and validated the model on the test dataset.

Using additional layers and increasing the number of hidden units improved the accuracy of the baseline MLP model for question quality assessment. Many experiments with different numbers of layers and hidden units are performed, and the model with hyperparameters given above gives the best results. The accuracy of the MLP-TF model is 78.0% which is 21.47% higher than the baseline model. Similarly, precision of 57.65%, recall of 57.65%, and *F*1-score of 57.65% is also much higher than the baseline MLP model.

A.4 Recurrent neural network

Recurrent neural network (RNN) [48,49] is a one-way directed network of neuron-like nodes. There are many variants of RNN, and the basic structure of RNN is shown in Figure A1. In Figure A1, the hidden state is denoted by s_t. It acts as a memory of the network and is capable of learning contextual information that is important for NLP classification tasks. The output at each step is calculated based on current input x_t and the memory s_t at time t. The main feature of an RNN is its hidden state, which captures sequential dependence in information.

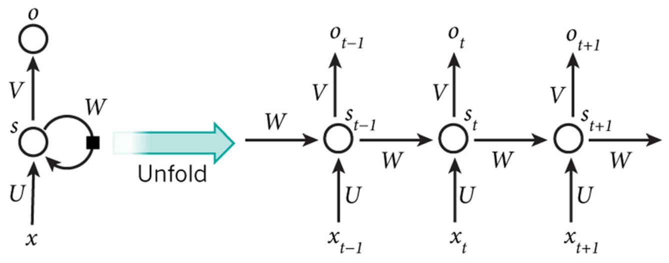


Figure A1: A recurrent neural network [Source: Nature].

In this research, two special kinds of RNN named long short term memory (LSTM) [50] networks and gated recurrent units (GRU) are used in experiments. LSTM is capable of remembering information over a long time period. GRU is also like LSTM, but forget gate is introduced in it and GRU has fewer parameters than LSTM. The units of LSTM and GRU are shown in Figure A2.

LSTM is implemented in the Google Colab environment [43]. The hyperparameters of LSTM and GRU are given in Table A1. The same architecture for LSTM and GRU is used, and a four-layer network is created. Input to both models is a features matrix. 100 units of LSTM and GRU are used in the first layer. The next two hidden dense layers consist of 50 units. The output layer consists of four units. The activation for LSTM, GRU, and

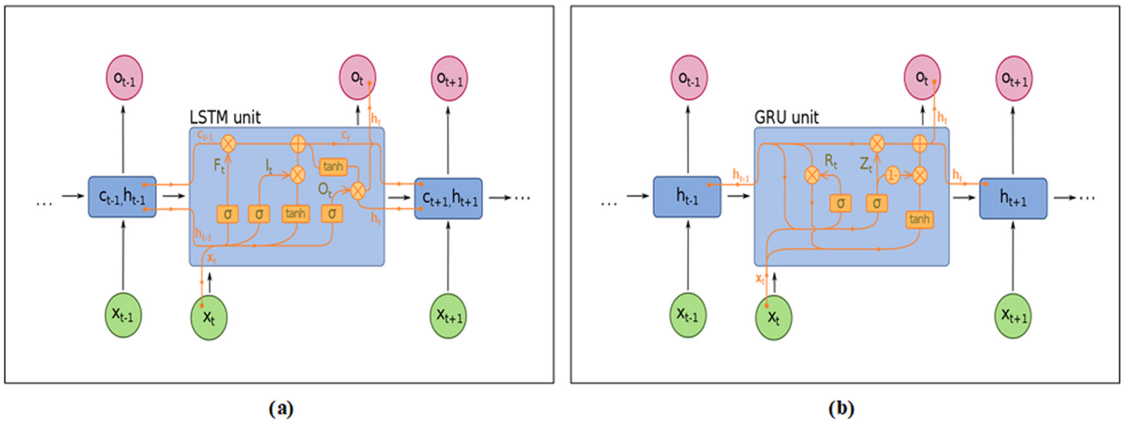


Figure A2: Structure of LSTM and GRU units [Source: Wikipedia].

Table A1: The control parameters values of LSTM & GRU

Parameter	Value
Model	LSTM, GRU
LSTM/GRU units	100
Dense Layers	2
Dense units per layer	50
Output units	4
Batch size	64
Epochs	50
Optimizer	Adam

dense layers is the Relu function. The Adam optimizer is used in the final layer. Data were divided into batch size of 64 and training was repeated for 50 epochs. The results of LSTM and GRU models are given in Table 7. The results of both models are almost the same. There is only a minor difference in precision, recall, and *F*1-score. The accuracy of both models is 78.56%. From the results, one can say that both models are equally good for classifying StackOverflow questions.

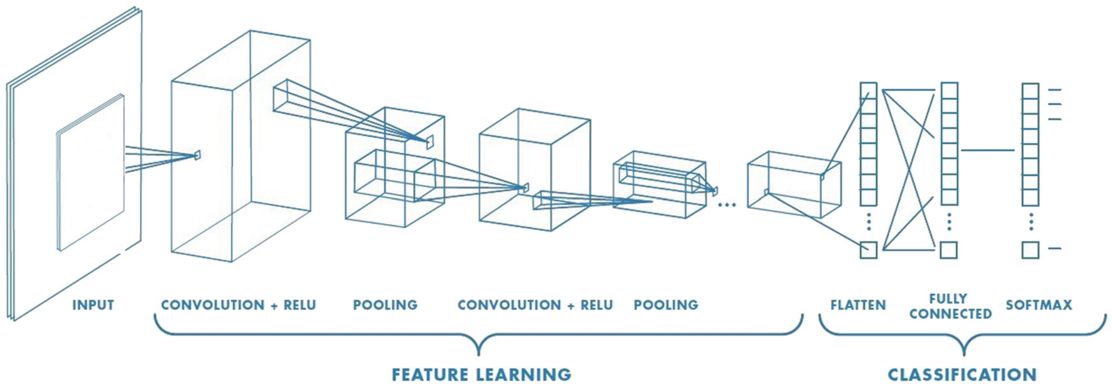


Figure A3: Architecture of CNN [53].

A.5 Convolution neural network

Convolutional neural networks (CNNs) [51,52] is a special type of deep neural network. CNN is composed of 1D or 2D convolution layers that can interpret special data. A sample architecture of CNN is shown in Figure A3 where convolution layers are followed by a pooling layer. The output of the convolution layer is flattened and then passed to dense layers. An advantage of CNN is that it can be used without pre-processing and can extract features from data. Features extraction is done by filters or kernels that are part of convolution layers.

CNN is implemented in Google Colab [43] by using Keras with Tensorflow backend. The architecture of the CNN model used to determine the quality of StackOverflow questions is given in Figure A4. Experiments with different numbers of convolution layers and hidden units are performed. The best results are achieved by the model as given in Figure A4. The model consists of two Conv1D layers, one flattened layer, one dense layer, and a dropout layer. Relu activation is used for hidden layers, and the softmax activation function is used in the

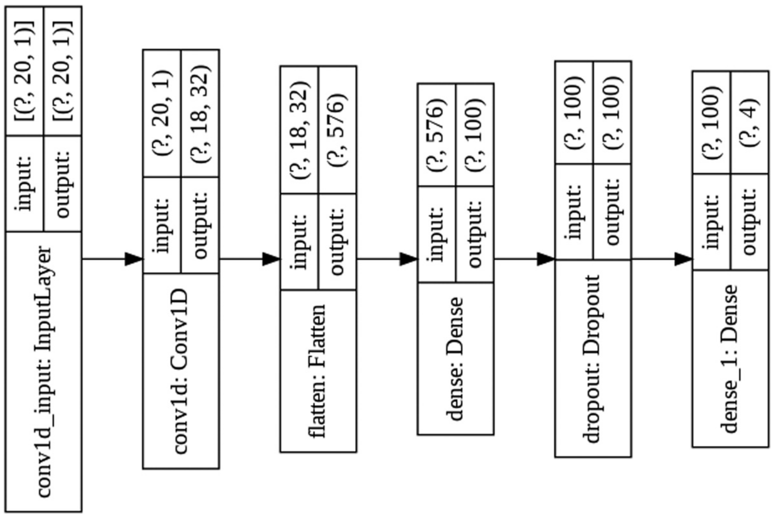


Figure A4: CNN model for stackoverflow question quality.

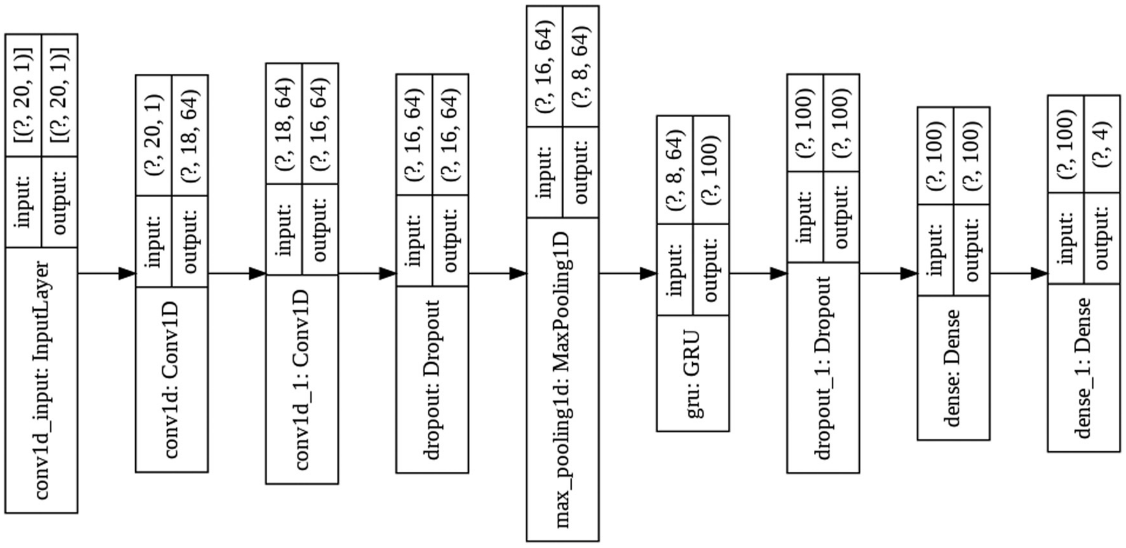


Figure A5: CNN + GRU model for StackOverflow question quality.

output layer. A dropout of 0.5 is added to avoid over-fitting. The data were divided into batch size of 64 and the training process was repeated for 50 epochs. The performance of the CNN model in classifying the StackOverflow questions is slightly better than the RNN models. The accuracy of 78.82%, precision of 57.65%, recall of 57.65%, and *F*1-score of 57.65% is achieved on an unseen test dataset. The results are given in Table 7.

A.6 Hybrid model of CNN and GRU

CNN+GRU is created in Google Colab [43] environment by using Keras with Tensorflow backend. The model consists of two Conv1D layers with 64 filters and 3 kernel size. After Conv1D layers, dropout of 0.5 is added to avoid over-fitting. The Maxpooling layer after the dropout is also added to down-sample input representation. The Maxpooling layer is followed by the GRU layer with 100 units. The dropout of 0.5 after the GRU layer is added to avoid over-fitting. The next layer is a dense layer with 100 units. All of these layers are activated by the relu activation function. The last layer is dense with 4 units which is the output layer. The output layer is activated by the softmax activation function. The model is compiled with Adam optimizer, and loss function is categorical cross-entropy. The architecture of the CNN+GRU model for StackOverflow questions quality assessment is given in Figure A5. The results of the CNN+GRU model are given in Table 7. The accuracy of the model on unseen testing data is 80.32% which is higher than the GRU and CNN models as implemented earlier.