International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

Comparative Analysis of Machine Learning and Deep Learning Approaches for Predicting Closed Questions on Stack Overflow

Puranasree M S[1], Rithanyavarshikaa M[2], Sowndarya [3], Swetha P[4], DR.D. Nithya[5]
[1,2,3,4]UG, Artificial Intelligence and Data Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore - 641043, Tamil Nadu, India.
[5]Associate Professor, CSE, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore - 641043, Tamil Nadu, India.
Emails ID: *puranasree1503@gmail.com[1], rithanyavarshikaa@gmail.com[2], sowndarya1027b@gmail.com[3] swethap4303@gmail.com[4], nithya_cse@avinuty.ac.in[5].*

Abstract
Stack Overflow, as a primary platform for programming-related knowledge sharing, faces ongoing challenges in maintaining content quality and managing duplicate questions. This research investigates two distinct computational approaches - Machine Learning and Deep Learning to predict question closure to enhance the efficiency of content question quality. The methodology encompasses two parallel approaches: an XGBoost classifier leveraging TF-IDF vectorization and a Convolutional Neural Network (CNN) architecture for semantic pattern recognition. The analysis utilizes a comprehensive dataset of labelled Stack Overflow questions, with both approaches incorporating text cleaning, tags removal and feature extraction in their respective pre-processing pipelines. Performance evaluation employs standard metrics including accuracy, precision, recall F1-score and confusion matrix. The comparative analysis provides insights into the relative strengths and limitations of traditional machine learning versus deep learning approaches, demonstrating each method's unique capabilities in identifying questions likely to be closed.

Keywords: *Convolutional Neural Network; Deep Learning; Machine Learning; Stack Overflow; XGBoost Classifier.*

1. Introduction
The rapid growth of online programming communities has transformed how developers share knowledge and seek solutions to technical challenges. Among these platforms, Stack Overflow stands out as a crucial resource for programmers of all experience levels. However, maintaining the quality of content on Stack Overflow remains a persistent challenge, as duplicate, off-topic, or low-quality questions can dilute the value of the platform. To address this issue, the platform employs a question closure system, where questions deemed unhelpful or redundant are flagged and removed from active circulation. Predicting which questions are likely to be closed can help streamline moderation efforts and improve the overall user experience. In a more specific application, Al-Ramahi et al. (2024) evaluated the effectiveness of deep neural networks in predicting Stack Overflow question quality. Their study compared neural networks with classical models such as Naïve Bayes, SVM, and Decision Trees, demonstrating that deep learning models outperformed traditional classifiers, achieving an accuracy of 80%. Their findings also highlighted the importance of network depth in classification performance, suggesting that optimization of hidden layers can further enhance predictive accuracy [1]. Hu and Yang (2024) advanced this field by comparing five machine learning models (decision trees, random forests, naive Bayes, support vector machines, and logistic regression) with two deep learning models (Bi-LSTM and BERT) for predicting Stack Overflow post quality. Their study found that

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

machine learning models performed within the 52%-74% accuracy range, while Bi-LSTM achieved 82%, and BERT reached a state-of-the-art accuracy of 92%. The study also emphasized the effectiveness of the attention mechanism in improving text classification tasks [2]. Arora et al. (2022) proposed, Ask It Right! Identifying Low-Quality Questions on Community Question Answering Services, introduce LQuaD, a multi-tiered hybrid model designed to detect low-quality questions on Stack Overflow. LQuaD utilizes transformers to determine semantic meaning and a graph convolutional network to link questions and tags, in contrast to traditional methods that rely on lexical, community-centric, and handcrafted features. On a dataset of 2.8 million questions, their model outperforms state-of-the-art methods by 21% in terms of F1-score. Survival analysis predicts the timelines for closing questions, enabling proactive intervention by users. Although effective, LQuaD's implementation of deep learning models increased computational costs and exhibited limited vocabulary analysis on diversing query texts [3]. Zhang and Chen (2019) in their study Duplicate Question Detection based on Neural Networks and Multi-head Attention, counter the limitation of employing a single neural network for Duplicate Question Detection (DQD) by introducing an ensemble learning strategy. Rather than sequentially combining networks susceptible to vanishing/exploding gradients and computational cost—they use parallel ensemble learning and regard different neural networks as separate learners. Their strategy includes recurrent and convolutional neural networks together with Multi-Head Attention to decrease correlation and performance gaps. Another new voting mechanism additionally improves accuracy at 89.3% on the Quora question pair's data set. Still, the computational intensity of the ensemble models restricted real-time usage, while performance on alternative data sets has not been evaluated [4]. Tóth et al. (2019) developed a deep learning-based NLP approach to predict question quality on Stack Overflow, achieving an accuracy of 74%. Their work primarily focused on linguistic characteristics of questions and employed Nesterov Stochastic Gradient Descent for classification. Similarly, Ruseti et al. (2018) assessed multiple recurrent neural network architectures, including GRU, BiGRU, and LSTM, achieving a peak accuracy of 81.22% in predicting question quality [5].

2. Methodology

The proposed system is designed to predict the likelihood of questions on Stack Overflow being closed, leveraging two distinct algorithms: XGBoost classifier and Convolutional Neural Network (CNN). These techniques reflect two distinct methods of machine learning and deep learning: XGBoost is a tree-based gradient boosting algorithm that is optimal for structured data while CNN is a deep learning algorithm that is ideal for processing unprocessed textual data and identifying semantic subtleties. In order to assess and contrast the performance of these algorithms and identify their individual advantages and disadvantages in automating question classification, the project will apply each method independently to the same dataset. The block diagram outlines the methodology, encompassing a parallel pipeline for both algorithms. The process begins with data collection, where question text, metadata (e.g., tags, creation date), and behavioural data (e.g., upvotes, downvotes, flags) are gathered. This raw data undergoes data pre-processing, which includes cleaning the text (removing noise, special characters and tags), tokenization, label encoding and normalization to ensure consistency across the dataset. In the feature extraction phase, techniques such as TF-IDF vectorization, Word2Vec, embeddings are employed to transform text into structured formats suitable for algorithmic processing. For the machine learning pipeline, the extracted numerical features are fed into the XGBoost classifier. XGBoost is a tree-based algorithm optimized for structured data, using an iterative process to improve predictions by minimizing a loss function. Meanwhile, for the deep learning pipeline, the processed text is fed into the CNN model, which utilizes convolutional layers to identify hierarchical patterns and capture semantic and contextual relationships in the text. [7]

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

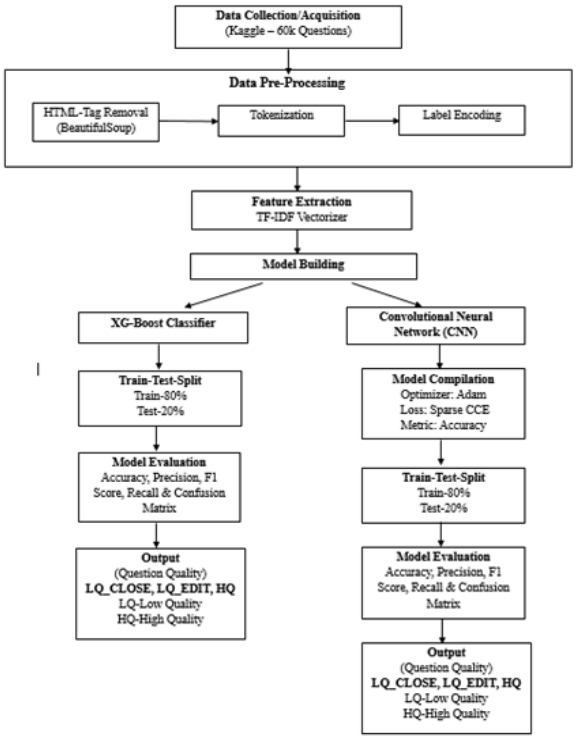

Figure 1 Block Diagram

2.1. Data Collection

The first step in the methodology is to gather data from Stack Overflow where a dataset from Kaggle containing 60,000 Stack Overflow questions, each labelled as "closed" or "non-closed" is used. Closed questions are flagged by moderators or community members due to issues like being off-topic, unclear, or duplicate, while non-closed questions meet quality standards. Figure 2 shows Data Collection.

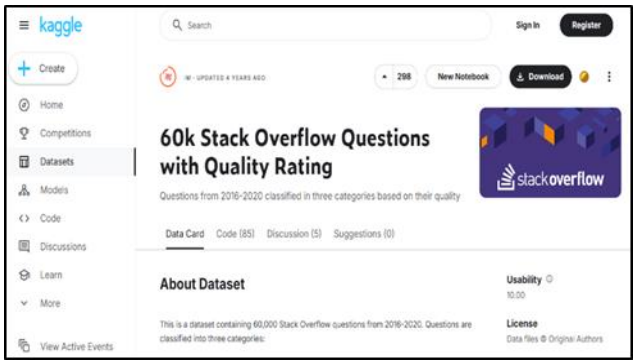

Figure 2 Data Collection

The dataset comprises of questions and associated metadata, including textual content (title and body), numerical features (votes, reputation, number of answers), and the target label indicating whether a question was closed (1) or remained open (0). Data acquisition is performed via the Stack Overflow API or web scraping tools such as Beautiful Soup. Figure 1 shows Block Diagram.

2.2. Data Pre-Processing

HTML Tag Removal

The dataset contains textual data in the form of body content, which may include HTML tags (e.g., , <p>, etc.). Since these HTML tags are not relevant to the classification task, they are removed to leave only the meaningful textual content. The removal of HTML tags is achieved using a parsing technique, ensuring that the data is clean and ready for analysis.

Label Encoding

The target variable, representing categories or labels (e.g., 'HQ', 'LQ_CLOSE', 'LQ_EDIT'), is initially in a categorical string format. To make the labels compatible with machine learning algorithms, they are converted into numeric values using label encoding. Each unique label is assigned a corresponding integer, which simplifies the model's learning process.

2.3. Feature Extraction

The textual data is vectorized using Term Frequency-Inverse Document Frequency (TF-IDF) to transform questions into numerical representations suitable for machine learning algorithms. Exploratory Data Analysis (EDA) is conducted to identify data distribution patterns, guiding feature engineering and selection. Additionally, statistical analyses such as word frequency distributions and correlation matrices are employed to gain deeper insights into feature significance.[6]

Representation using word2vec

Text Vectorization Machine learning algorithms, including neural networks, require numerical data to perform computations. Since the dataset consists of textual data, it is transformed into a numerical format using TF-IDF vectorization. This technique evaluates the importance of each word in a document relative to the entire corpus, effectively converting the raw text into feature vectors. By applying this method, the model can handle word frequency and its significance across all documents.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

Transformation Process TF-IDF involves two components: Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures the frequency of a word in a given document, while IDF measures the importance of the word in the entire corpus. The combination of these two metrics provides a weighted representation of each word's significance.

2.4. Dataset Splitting

The dataset is divided into two subsets: a training set and a testing set. This division is crucial to assess the model's performance on unseen data. Typically, 80% of the data is used for training the model, and the remaining 20% is set aside for evaluation.

Table 1 Training and Testing split

Question Type	Number of Questions in Training Data	Number of Questions in Test Data
LQ_EDIT	16,000	4,000
LQ_CLOSE	16,000	4,000
HQ	16,000	4,000

This ensures that the model learns from one portion of the data and is tested on another to check for generalization. Table 1 shows Training and Testing split.

2.5. Model Development

2.5.1 XGBoost Classifier

The machine learning approach employs XGBoost classifier, a gradient boosting algorithm optimized for classification tasks. XGBoost is selected due to its efficiency, ability to handle missing values, and feature importance ranking capabilities. The model is trained using Scikit-learn, leveraging its gradient boosting framework to improve classification accuracy. The model undergoes iterative training cycles, where decision trees are sequentially added to correct previous errors., The TF-IDF vectors serve as input features for the XGBoost model, which learns patterns associated with question closure based on structured and unstructured data. Hyperparameter tuning is conducted using Grid Search and Randomized Search techniques to optimize learning rate, maximum depth, and number of estimators, ensuring optimal model performance. Regularization techniques such as L1 and L2 penalties are applied to prevent over fitting.

2.5.2 Convolutional Neural Networks (CNN)

CNNs are deep learning models that are particularly effective for image and text data. They are designed to automatically learn spatial hierarchies of features. In this project, CNNs are used to process the textual data, learning patterns in the sequences of words that may indicate whether a question is likely to be closed.

- **Input Layer:** The model accepts the TF-IDF vectors as input, where each word is represented by a numerical value reflecting its importance in the document.
- **Hidden Layers:** The network contains one or more hidden layers, which use activation functions such as ReLU (Rectified Linear Unit) to introduce non-linearity and allow the model to learn complex patterns in the data.
- **Output Layer:** The output layer consists of as many nodes as the number of classes (in this case, three categories: 'HQ', 'LQ_CLOSE', 'LQ_EDIT'). A softmax activation function is applied to ensure the output represents probabilities for each class.[4]

Working of CNN:

- **Convolutional Layers:** These layers apply filters (kernels) to the input text, capturing local patterns or features. The filters slide over the input data (in this case, the text) and produce feature maps that highlight important characteristics.
- **Activation Function (ReLU):** The ReLU (Rectified Linear Unit) function is applied to introduce non-linearity into the network, enabling it to learn complex relationships.
- **Pooling Layers:** Pooling (typically max pooling) reduces the dimensionality of the data and extracts the most important

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

features from the feature maps. This helps prevent overfitting and reduces computational complexity.

- **Fully Connected Layers:** After the convolution and pooling layers, the CNN typically includes fully connected layers (dense layers) to combine the learned features and make the final classification decision.

In this project, the CNN will be used to analyze the content of the questions by processing the textual features (question title and body). The model will learn to recognize patterns such as vague wording, use of inappropriate language, or other factors that could lead to question closure.

2.6. Model Training

The model is trained using the training dataset, and during training, the weights of the neural network are adjusted based on the input data. The model is optimized using an appropriate loss function (sparse categorical cross-entropy) and an optimizer (Adam), which minimizes the prediction error. A validation set is used to monitor the model's performance and prevent overfitting.

2.6.1 Classification Report

A classification report is generated to provide a detailed view of the model's performance across each class. This includes key metrics like precision, recall, and F1-score, which are essential for evaluating the model's ability to distinguish between different categories. Figure 3 shows Training Output of XGBoost Classifier.

- **Accuracy:** Measures the overall correctness of the model.
- **Precision:** Assesses how many predicted closed questions were actually closed.
- **Recall:** Evaluates the model's ability to identify closed questions correctly.
- **F1-score:** Provides a balance between precision and recall, especially in cases of class imbalance. The deep learning model achieved 98% accuracy, compared to 92% accuracy for the machine learning model. It also demonstrated superior precision, recall, and F1-score across all categories (HQ, LQ_CLOSE, LQ_EDIT).

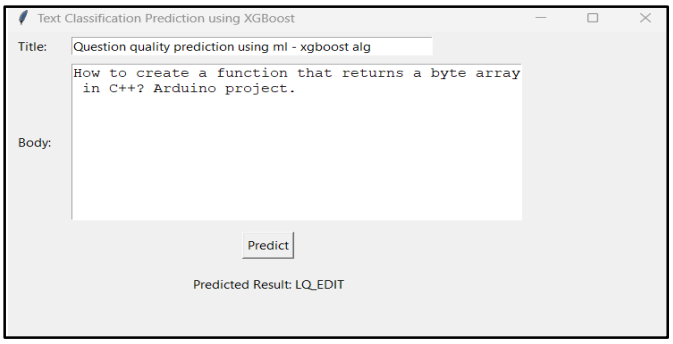

Figure 3 Training Output of XGBoost Classifier

2.6.2 Confusion Matrix:

A confusion matrix is produced to visualize how well the model performs across all classes. This matrix displays the true positives, false positives, true negatives, and false negatives, helping to identify which classes the model struggles with**.** Figure 4 shows Training Output of CNN Algorithm.

```
Classification Report:
              precision    recall  f1-score   support

          HQ       0.99      0.98      0.98       264
    LQ_CLOSE       0.95      0.98      0.97       174
     LQ_EDIT       0.98      0.97      0.97       176

    accuracy                           0.98       614
   macro avg       0.97      0.98      0.97       614
weighted avg       0.98      0.98      0.98       614

Confusion Matrix:
 [[258   3   3]
 [  3 171   0]
 [  0   6 170]]
```

Figure 4 Training Output of CNN Algorithm

```
Model Accuracy: 0.92
Classification Report:
              precision    recall  f1-score   support

          HQ       0.93      0.97      0.95       534
    LQ_CLOSE       0.91      0.91      0.91       372
     LQ_EDIT       0.91      0.86      0.88       322

    accuracy                           0.92      1228
   macro avg       0.92      0.91      0.92      1228
weighted avg       0.92      0.92      0.92      1228

Confusion Matrix:
 [[516  12   6]
 [ 12 339  21]
 [ 24  21 277]]
```

Figure 5 Testing Output of XGBoost Classifier

2.7. Model Evaluation

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

After training, the model is evaluated on the test set to measure its performance. The accuracy metric is calculated to assess the percentage of correct predictions made by the model. Performance evaluation is conducted using standard classification metrics, including: The testing results are shown in Figures 5 and 6. The quality of the questions can be classified into three types: LQ_CLOSE, LQ_EDIT, and HQ_CLOSE. The LQ_CLOSE category contains low-quality questions that include an answer, whereas the HQ_CLOSE category has high-quality questions with comparable answers. Finally, the LQ_EDIT category contains questions that have an answer but are marked incorrect.[8]

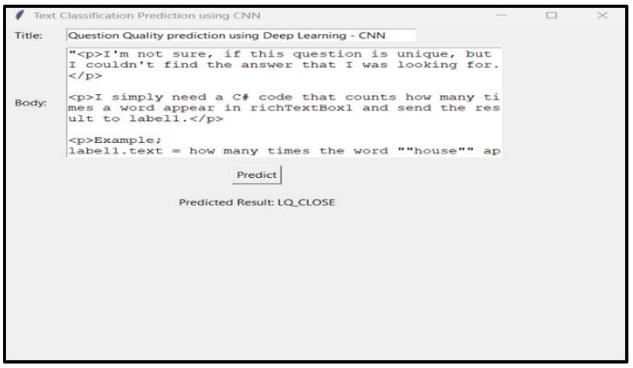

Figure 6 Testing Output of CNN Algorithm

3. Results and Discussion

3.1. Graphical Analysis

The comparative performance of Machine Learning (XGBoost) and Deep Learning (CNN) models in predicting closed questions on Stack Overflow indicates significant differences in their performance. Figure 8 shows Comparative Graphical Analysis of Machine Learning and Deep Learning using Line chart.

3.2. Discussion

CNN outperforms XGBoost in question categorization mostly because to its ability to learn hierarchical textual characteristics. This advantage is demonstrated by several significant findings: With higher recall and F1-scores for various question types, CNN demonstrates superior generalization, demonstrating a deeper comprehension of context and semantics. Notably, CNN outperforms XGBoost in processing noisy edited questions, demonstrating more pattern recognition in dirty data overall with much higher recall and F1-scores. Additionally, CNN lowers false positives, increasing precision - a critical component of accurate content management. Table 2 shows Performance analysis of Machine Learning and Deep Learning.

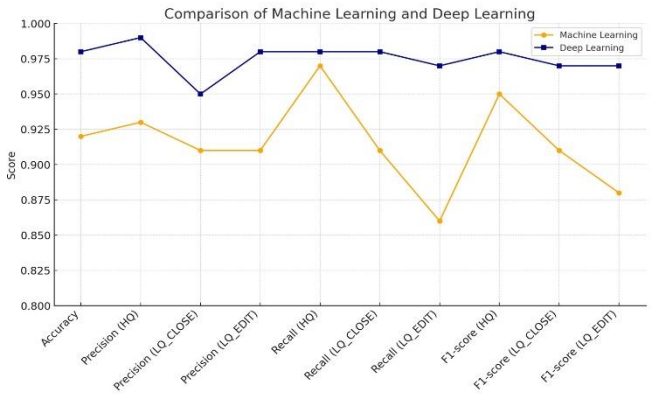

Figure 8 Comparative Graphical Analysis of Machine Learning and Deep Learning using Line chart

Table 2 Performance analysis of Machine Learning and Deep Learning

Metric Name	Machine Learning	Deep Learning
Accuracy	0.92	0.98
Precision (HQ)	0.93	0.97
Precision (LQ_CLOSE)	0.91	0.96
Precision (LQ_EDIT)	0.90	0.95
Recall (HQ)	0.96	0.97
Recall (LQ_CLOSE)	0.89	0.94
Recall (LQ_EDIT)	0.80	0.93
F1-score (HQ)	0.94	0.96
F1-score (LQ_CLOSE)	0.87	0.94
F1-score (LQ_EDIT)	0.83	0.93

Conclusion

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0106
e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 647-653

This research addresses preserving Stack Overflow content quality by comparing machine learning and deep learning for predicting closed questions. We examined two approaches: a CNN for semantic pattern recognition and XGBoost with TF-IDF vectorization. Results show CNNs outperform XGBoost, improving content moderation on massive sites. The proposed technique reduces manual moderation needs and provides a scalable way to maintain content quality. Future research should address dataset representativeness, computational complexity, and deep learning model interpretability. This work highlights advanced computational methods for content moderation and provides insights on deep learning versus conventional machine learning advantages for question classification.

Future Scope

The research presented here shows that how different learning techniques are successful at forecasting Stack Overflow question closures. However, there is still a great of potential for further study and advancement. One potential direction is the exploration of additional classification algorithms to further diversify the ensemble and improve prediction accuracy. The integration of graph-based algorithms to capture intricate linkages between questions, tags, and user interactions may be the main focus of future research. Furthermore, by combining the strengths of different architectures, such as Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs), or Transformer-based models, ensembles of neural networks could be created to improve prediction accuracy by utilizing a variety of feature extraction capabilities. Other important areas of focus will be addressing class imbalance and dataset representativeness, as well as improving the models' interpretability and scalability. These developments could further improve automated content moderation systems and improve user experience on large-scale platforms.

References

[1]. Mohammad Al-Ramahi, Izzat Alsmadi, Abdullah Wahbeh, "Predicting Question Quality on StackOverflow with Neural Networks," 2024, abs/2404.14449 [cs.CL], 20 April 2024.
[2]. J. Hu and B. Yang, "Posts Quality Prediction for StackOverflow Website," in IEEE Access, vol. 12, pp. 135601-135615, 2024, doi: 10.1109/ACCESS.2024.3440879.
[3]. U. Arora, N. Goyal, A. Goel, N. Sachdeva and P. Kumaraguru, "Ask It Right! Identifying Low-Quality questions on Community Question Answering Services," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022.
[4]. Heng Zhang, Liangyu Chen, "Duplicate Question Detection Based on Neural Networks and Multi-head Attention," 2019 International Conference on Asian Language Processing (IALP).
[5]. 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. 14th International Conference on Software Technologies, 2019.
[6]. Kumar, R., & Gautam, S. (2023). Quality prediction of Stack Overflow questions using transformer-based models. Journal of Software Engineering and Applications.
[7]. Srikant Kumar, Anjali Mohapatra, Sabyasachi Patra, Sunakshi Mamgain, "Detection of Intent-Matched Questions Using Machine Learning and Deep Learning Techniques," 2019 International Conference on Information Technology (ICIT).
[8]. Zhifang Liao, Wenlong Li, Yan Zhang, Song Yu, "Detecting Duplicate Questions in Stack Overflow via Semantic and Relevance Approaches," 2021 28th Asia-Pacific Software Engineering Conference (APSEC).