Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 2: 2025
ISSN : **1906-9685**

Improving Content Curation: Prediciting Post Quality On Stack Over Flow

#1 Sk.SALMA, #2 K.JAYA KRISHNA

#1 MCA Scholar

#2 Assistant Professor

DEPARTMENT OF MASTER OF APPLICATIONS

QIS COLLEGE OF ENGINEERING AND TECHNOLOGY

Vengamukkapalem(V), Ongole, Prakasam dist., Andhra Pradesh- 523272

Abstract: The large volume of user-generated material on sites like StackOverflow necessitates effective question quality evaluation because poor or irrelevant queries might strain processing power. The detection and classification of question quality are improved by combining machine and deep learning methods, such as Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes, BERT, and BI-LSTM. The analysis is guided by performance evaluation criteria like F-score, recall, accuracy, and precision. Several word embedding methods are investigated, including Word2Vec, Doc2Vec, and TF-IDF; Doc2Vec produces better features by encapsulating the semantic meaning of words. The algorithms are trained and tested on the Stack Overflow dataset, which consists of question text and quality labels classified as 'HQ' (High Quality), 'LQE' (Low Quality Edit), and 'LQ Close (LQ Close). Remarkably, the BERT model attains a 93% accuracy rate, whilst the novel CNN2D model exhibits a remarkable 94% accuracy rate, indicating the possibility of enhanced feature optimization via multi-dimensional array processing. The results highlight how crucial cutting-edge methods are for improving prediction accuracy and helping to effectively monitor the caliber of questions on online platforms.

Index Terms—StackOverflow, Question Quality Assessment, Machine Learning, Deep Learning, BERT, Bi-LSTM, CNN2D, Doc2Vec, Text Classification

1. INTRODUCTION

The growth of professional knowledge-sharing platforms like StackOverflow has made it easier for developers, learners, and experts to exchange technical knowledge. However, the increasing

number of user-generated questions has also led to a surge in low-quality or irrelevant content. These questions can overwhelm both the system and the community, making it difficult for users to find useful answers and for moderators to maintain platform quality. Efficient question quality assessment is therefore essential to ensure smooth knowledge sharing and enhance the overall user experience.

Traditional methods of evaluating question quality often rely on manual moderation or simple machine learning techniques, which may not capture the complexities of question text and context. To address this, advanced machine learning and deep learning algorithms, including Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes, BERT, and Bi-LSTM, can be employed to automatically classify questions based on quality. These methods help identify patterns in text data and improve prediction accuracy.

Furthermore, the use of advanced word embedding techniques like Doc2Vec, Word2Vec, and TF-IDF allows the system to capture semantic relationships between words, with Doc2Vec providing superior feature representation. In addition, 2D Convolutional Neural Networks (CNN2D) are utilized for optimized feature extraction, enhancing the classification performance. By integrating these techniques, the proposed system aims to reduce the presence of low-quality questions and create a more effective and efficient question-answering environment on StackOverflow.

2. LITERATURE SURVEY

2.1 Do Subjectivity and Objectivity Always Agree? A Case Study with Stack Overflow Questions:

https://ieeexplore.ieee.org/abstract/document/10174053

ABSTRACT: Users on Stack Overflow (SO) use a voting system to rate the quality of articles (questions and replies). People typically use the net votes (upvotes minus downvotes) that a post gets as a rough measure of its quality. But nearly half of the questions that had working answers got more downvotes than upvotes. Also, roughly 18% of the recognized answers (i.e., verified solutions) don't get the most votes either. All of these surprising results make me question the trustworthiness of the evaluation system used at SO. A lot of users also don't like the evaluation, especially when their postings get downvotes. Thus, thorough validation of the subjective evaluation is essential to guarantee an impartial and dependable quality assessment system. This article compares the subjective evaluation of questions with their objective evaluation, utilizing 2.5 million questions and ten text analysis measures. Our analysis reveals that four objective indicators align with the subjective rating, two do not align, one is ambiguous, and the other three neither confirm nor refute the subjective evaluation. After that, we create machine learning models to sort the inquiries that were encouraged and those that were not. With a maximum accuracy of roughly 76%–87%, our models do better than the best models out there.

2.2 Is this question going to be closed? Answering question closibility on Stack Exchange:

https://journals.sagepub.com/doi/abs/10.1177/01655515221118665

ABSTRACT: There are a lot of inquiries on community question answering sites (CQAs) that

never get answered. To deal with the issue, Stack Exchange now lets experienced users designate new questions as closed if they aren't very good. A question can't get answers once it's closed. But it takes time to find and close bad questions. The aim of this article is to create a supervised machine learning system that forecasts question closability, which is the likelihood that a freshly posted question will be closed. The supervised machine learning system builds on existing research on the quality of CQA questions. It includes 17 features that are divided into four groups: asker features, community features, question content features, and textual features. We examined how well the built method worked by using questions from Stack Exchange on 11 randomly chosen themes. The categorization performance was mostly good and better than the baseline. No matter what the questions were about, most of the measures of precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC) were above 0.90. By formulating the idea of question closibility, the article advances prior CQA research concerning question quality. This study empirically examines query closibility across 11 randomly selected themes, in contrast to prior research that mostly focused on programming-related inquiries from Stack Overflow. The set of criteria utilized for classification provides a framework for question closibility that is both more broad and more economical than previous studies.

2.3 Support-BERT: Predicting Quality of Question-Answer Pairs in MSDN using Deep Bidirectional Transformer:

https://arxiv.org/abs/2005.08294

ABSTRACT: It is hard to say what makes a good question or answer on community support sites like Microsoft Developers Network, Stackoverflow, Github, and others. It is even harder to make a prediction model for good questions and replies. Previous studies have examined question quality models and response quality models independently, utilizing meta-properties such as the amount of up-votes, the credibility of the individual posting the questions or answers, the titles of the posts, and context-agnostic natural language processing features. Nonetheless, the research lacks a cohesive question-answer quality paradigm for community question answering platforms. In this concise study, we address the quality Q&A modeling challenges from community help websites by employing a recently created deep learning model with bidirectional transformers. We examine the feasibility of transfer learning in Q&A quality modeling by utilizing Bidirectional Encoder Representations from Transformers (BERT) trained on distinct tasks derived from Wikipedia. A further pre-training of the BERT model, together with fine-tuning on Q&As sourced from the Microsoft Developer Network (MSDN), can enhance automated quality prediction performance to over 80%. In addition, the implementations are done to use AzureML in the Azure knowledge base system to deliver the finetuned model in real time.

2.4 Asking Questions is Easy, Asking Great Questions is Hard: Constructing Effective Stack Overflow Questions:

https://digitalcommons.oberlin.edu/honors/694/

ABSTRACT: This paper examines and aims to enhance the methods by which Stack Overflow question posts can generate responses. We identify three critical elements prevalent in numerous prior successful or responsive queries using statistical data analysis and literature reviews. Next, we show a sample sidebar for the ask page that uses these factors

to (1) dynamically rate the quality of questions in construction, (2) show answer previews of relevant questions, and (3) help the identified factors to new askers as they work on their questions.

2.5 Predicting closed questions on community question answering sites using convolutional neural network:

https://link.springer.com/article/10.1007/s00521-019-04592-0

ABSTRACT: Every day, community question-and-answer sites get a lot of queries and replies. It has been noticed that the site moderators have marked a number of questions as closed. These kinds of inquiries make things harder for the moderators and make users unhappy. The goal of this work is to guess whether a question that has just been posted will be marked as closed in the future and to suggest a possible reason for why it was closed. There are two models: (1) a baseline model that uses traditional machine learning methods and (2) deep learning models like convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. These models sort a question into one of five groups: (1) open, (2) off-topic, (3) not a real question, (4) too constructive, or (5) too localized. The baseline approach necessitates handcrafted features, hence failing to maintain semantics. But CNN and LSTM networks may keep the meaning of the words in a question and use several hidden layers to find hidden features in the text. The LSTM network works better than CNN and other classic machine learning models. The proposed model can be used as a first step to screen the closed question when it is posted, which made things easier for site moderators. As far as we know, this is the first time someone has predicted the closed question and why it will be closed.

3. METHODOLOGY

The methodology involves collecting and preprocessing StackOverflow questions labeled as 'HQ', 'LQE', and 'LQ Close', including text cleaning, tokenization, and stop word removal. Features are extracted using word embedding techniques like TF-IDF, Word2Vec, and Doc2Vec, with Doc2Vec providing superior semantic representation. Both machine learning algorithms (Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes) and deep learning models (BERT, Bi-LSTM, CNN2D) are applied to classify question quality, capturing complex patterns and contextual information. CNN2D further optimizes feature extraction using multi-dimensional arrays. Model performance is evaluated using accuracy, precision, recall, and F-score to identify the most effective approach for automating question quality assessment on StackOverflow.

A. Proposed Work:

The proposed work focuses on enhancing question quality prediction on StackOverflow by extending traditional machine learning and deep learning approaches with advanced Convolutional Neural Network techniques. Specifically, CNN2D and CNN3D models are employed alongside standard algorithms like Random Forest, SVM, Naïve Bayes, and Bi-LSTM. These convolutional networks process multi-dimensional arrays of question text features, allowing for optimized feature extraction that captures complex patterns and contextual relationships more effectively than one-dimensional models. By leveraging this extension, the system aims to improve the classification accuracy of high-

quality versus low-quality questions and reduce misclassification.

Furthermore, the system integrates advanced word embedding techniques, particularly Doc2Vec, to transform textual data into rich semantic vectors that enhance model understanding. The combination of multi-dimensional CNN architectures with powerful embeddings allows for a more nuanced representation of question content. This approach not only improves predictive performance but also ensures faster and more reliable assessment of incoming questions, contributing to better content moderation, efficient user engagement, and overall improvement of knowledge-sharing quality on the StackOverflow platform.

B. System Architecture:

The system architecture for the Posts Quality Prediction on StackOverflow consists of four main components: data collection, preprocessing, feature extraction, and classification. First, the dataset containing StackOverflow questions and their quality labels (‘HQ’, ‘LQE’, ‘LQ Close’) is collected. Next, preprocessing is performed, including text cleaning, tokenization, and removal of stop words, to prepare the data for analysis. In the feature extraction phase, advanced word embedding techniques such as TF-IDF, Word2Vec, and Doc2Vec are used to convert textual data into numerical vectors that capture semantic meaning. Finally, the classification module applies machine learning algorithms (Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes) and deep learning models (BERT, Bi-LSTM, CNN2D) to predict question quality. The system evaluates model performance using metrics like accuracy, precision, recall, and F-score, allowing the most effective algorithms to be identified and implemented for automated question quality assessment.

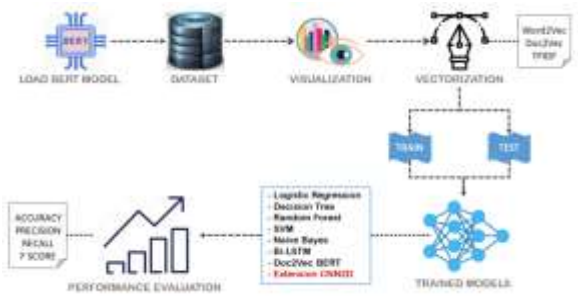

Fig1. proposed architecture

C. MODULES:

i. **Data Collection**

- Gather StackOverflow questions along with quality labels (‘HQ’, ‘LQE’, ‘LQ Close’) from the dataset.
- Ensure the dataset is complete and suitable for preprocessing and analysis.

ii. Data Preprocessing

- Perform text cleaning, tokenization, and removal of stop words.
- Handle duplicates, punctuation, and irrelevant characters to prepare data for modeling.

iii. Feature Extraction

- Apply word embedding techniques such as TF-IDF, Word2Vec, and Doc2Vec.
- Convert question text into numerical vectors capturing semantic and contextual information.

iv. Question Classification

- Implement machine learning algorithms: Logistic Regression, Decision Tree, Random Forest, SVM, and Naïve Bayes.

- Apply deep learning models: BERT, Bi-LSTM, and CNN2D for advanced feature learning and context understanding.

v. Model Evaluation

- Measure performance using accuracy, precision, recall, and F-score.
- Compare different models and embedding techniques to select the most effective solution.

vi. Prediction & Deployment

- Predict the quality of new questions in real-time.
- Integrate the system for automated content management on StackOverflow.

D. Algorithms:

a) Logistic Regression

Logistic Regression is a supervised learning algorithm used for binary or multi-class classification. It models the probability of question quality labels based on input features and is often considered the starting point for many text classification problems due to its simplicity and interpretability. It is highly effective when the relationship between features and output is approximately linear.

b) Decision Tree

Decision Tree is a tree-based model that classifies questions by creating decision rules derived from dataset features. It provides clear visualization of decision paths, making it easier to understand why a question is classified as high or low quality. Decision Trees are also fast to train and can handle both numerical and categorical data efficiently.

c) Random Forest

Random Forest is an ensemble of multiple decision trees. It improves classification accuracy and reduces overfitting through majority voting. Its robustness to noise and ability to handle large datasets make it ideal for analyzing large-scale StackOverflow questions. Additionally, it can provide feature importance scores, helping to understand which question features influence quality the most.

d) Support Vector Machine (SVM)

SVM finds optimal hyperplanes that separate different question quality categories. It is especially effective in high-dimensional spaces, such as text embeddings, and can work with both linear and non-linear kernel functions. SVM is known for its strong generalization capability, which helps in predicting the quality of unseen questions accurately.

e) Naïve Bayes

Naïve Bayes is a probabilistic classifier that predicts question quality based on the likelihood of feature occurrences. Despite its "naïve" assumption of feature independence, it often performs surprisingly well in text classification tasks. It is fast, computationally efficient, and works well with large vocabularies typical of question datasets.

f) Bi-LSTM (Bidirectional Long Short-Term Memory)

Bi-LSTM processes sequences in both forward and backward directions, capturing contextual dependencies in question text. It is excellent for understanding semantics where the meaning of a word depends on surrounding words. Bi-LSTM can remember long-term dependencies, which is useful for analyzing complex, multi-sentence questions.

g) BERT (Bidirectional Encoder Representations from Transformers)

BERT is a pre-trained transformer model that captures deep contextual relationships between words. It understands both left and right context simultaneously, making it highly effective for nuanced text interpretation. BERT can be fine-tuned with StackOverflow questions, allowing it to adapt to domain-specific language patterns and improve prediction accuracy.

h) Doc2Vec

Doc2Vec is a word embedding technique that converts text into fixed-length vectors while preserving semantic meaning. It goes beyond simple bag-of-words approaches, capturing the relationships between words and the context of entire documents. This helps in representing question content more accurately for machine learning and deep learning models.

i) CNN2D (Convolutional Neural Network)

CNN2D processes multi-dimensional arrays of text features using convolutional layers. It excels at detecting local patterns and hierarchies in data, such as key phrases or word combinations that indicate question quality. CNN2D is also efficient with large datasets and can significantly improve classification performance by learning complex feature interactions.

4. EXPERIMENTAL RESULTS

The proposed system was evaluated using a StackOverflow dataset containing questions labeled as 'HQ', 'LQE', and 'LQ Close'. Different word embedding techniques—TF-IDF, Word2Vec, and Doc2Vec—were applied to extract features, with Doc2Vec providing the best semantic representation. Multiple machine learning algorithms (Logistic Regression, Decision Tree, Random Forest, SVM, Naïve Bayes) and deep learning models (BERT, Bi-LSTM, CNN2D) were trained and tested for question quality classification.

The evaluation metrics included accuracy, precision, recall, and F-score. Among the models, BERT achieved an accuracy of 93%, while the CNN2D model outperformed all others with an accuracy of 94%. The results indicate that CNN2D, combined with Doc2Vec embeddings, effectively captures complex patterns and semantic information in question text, leading to superior classification performance. These findings demonstrate the effectiveness of the extended approach in automating quality assessment and improving the overall management of questions on StackOverflow.

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

$$Accuracy = \frac{(TN + TP)}{T}$$

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1 = 2 \cdot \frac{(Recall \cdot \Pr e\,cision)}{(Recall + \Pr e\,cision)}$$

Precision: Precision evaluates the fraction of correctly classified instances or samples among the

ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\Pr e\,cision = \frac{TP}{(TP + FP)}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{(FN + TP)}$$

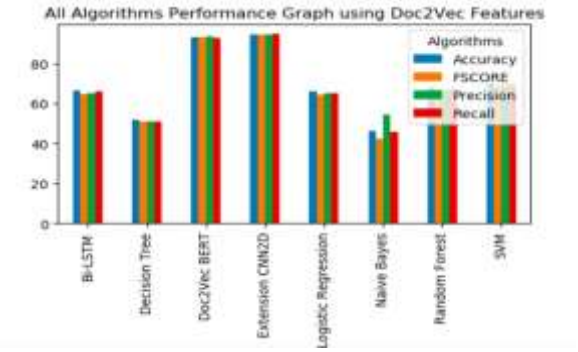

Fig 2.performance evaluation

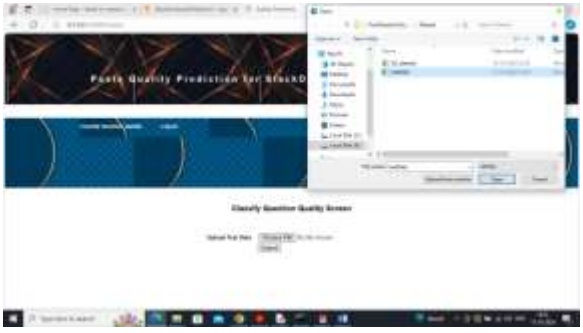

Fig 3.Upload dataset

Fig.4.. predicted results

5. CONCLUSION

In conclusion, the suggested system greatly improves the ability to find out how good questions are on sites like StackOverflow by using a wide range of machine and deep learning algorithms, such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), Naïve Bayes, BERT, and BI-LSTM. The system uses a dataset of Stack Overflow questions to focus on successful classification. It does this by using advanced word embedding techniques including Word2Vec, Doc2Vec, and TF-IDF, with Doc2Vec being the best at representing meaning. Also, the use of 2D Convolutional Neural Networks (CNN) makes feature extraction more efficient, which helps increase classification accuracy. The proposed methods not only try to find bad questions, but they also try to make the whole question-and-answer process work better. The algorithms' performance evaluation shows a huge increase in accuracy, with BERT getting 93% and the CNN model getting 94%. This means that the suggested approach can provide reliable and efficient quality identification, which will make it easier and more effective for people to share information on StackOverflow and other similar sites. The research findings emphasize the necessity of employing modern algorithms and

strategies to enhance the quality of material in online question-and-answer forums.

6. FUTURE SCOPE

The future scope of this research involves augmenting the dataset to include a broader array of question-and-answer platforms, hence improving the model's generalizability across many disciplines. In addition, using sentiment analysis could help us understand how engaged and satisfied users are with the quality of the questions. Investigating hybrid models that include conventional and deep learning methodologies may enhance classification precision. Adding the ability to process data in real time could make it possible to assess quality dynamically as questions are asked. Finally, adding ways for users to give input could help the model get better over time, making sure that the system keeps up with changes in user questions and content quality standards.

REFERENCES

[1] S. Mondal, M. M. Rahman, and C. K. Roy, ‘‘Do subjectivity and objectivity always agree? A case study with stack overflow questions,’’ 2023, arXiv: 2304.03563.

[2] P.K.Roy, J.P.Singh, and S.Banerjee, ‘‘Is this question going to be closed? Answering question closibility on stack exchange,’’ J. Inf. Sci., Oct. 2022, Art. No. 016555152211186, doi: 10.1177/01655515221118665.

[3] B. Sen, N. Gopal, and X. Xue, ‘‘Support-BERT: Predicting quality of question-answer pairs in MSDN using deep bidirectional transformer,’’ 2020, arXiv: 2005.08294.

[4] Hsieh, J. W. (2020). Asking questions is easy, asking great questions is hard: Constructing Effective Stack Overflow Questions.

[5] P. K. Roy and J. P. Singh, ‘‘Predicting closed questions on community question answering sites using convolutional neural network,’’ Neural Comput. Appl., vol. 32, no. 14, pp. 10555–10572, Jul. 2020, doi: 10.1007/s00521-019-04592-0.

[6] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, and Z. Xiong, ‘‘Question/answer matching for CQA system via combining lexical and sequential information,’’ in Proc. AAAI Conf. Artif.Intell., 2015, pp. 275–281. Accessed: Feb. 22, 2024.[Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9178

[7] YueLiu, A. Tang, FeiCai, P. Ren, and Z. Sun, ‘‘Multi-feature based question–answerer model matching for predicting response time in CQA,’’ Knowl.-Based Syst., vol. 182, Oct. 2019, Art. no. 104794. 135614

[8] H. Fan, Z. Ma, H. Li, D. Wang, and J. Liu, ‘‘Enhanced answer selection in CQA using multi-dimensional features combination,’’ Tsinghua Sci. Technol., vol. 24, no. 3, pp. 346–359, Jun. 2019.

[9] A. Baltadzhieva and G. Chrupała, ‘‘Question quality in community question answering forums: A survey,’’ ACM SIGKDD Explorations Newslett., vol. 17, no. 1, pp. 8–13, Sep. 2015, doi: 10.1145/2830544. 2830547.

[10] X. Shi, Z. Chen, H. Wang, D.-Y.Yeung, W.-K. Wong, and W. Woo, ‘‘Convolutional LSTM network: A machine learning approach for precipitation nowcasting,’’ in Proc. Adv. Neural Inf.

Process. Syst., vol. 28, 2015, pp. 1–9. Accessed: Feb. 23, 2024.[Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/07563a3fe3bbe7e3ba8443 1ad9d055af-Abstract.html

[11] M.Ellmannand, M.Schnecke, ‘‘Two perspectives on software documentation quality in stack overflow,’’ in Proc. 4th ACM SIGSOFT Int. Workshop NLP Softw. Eng. Lake Buena Vista, FL, USA: ACM, Nov. 2018, pp. 6–9, doi: 10.1145/3283812.3283816.

[12] Y.Yao, H.Tong, T.Xie, L.Akoglu, F.Xu, and J.Lu, ‘‘Want a good answer? Ask a good question first!’’ 2013, arXiv: 1311.6876.

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. (2017). Automatic Differentiation in PyTorch.Accessed: May 21, 2024.[Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ

[14] P. Braslavski, D. Savenkov, E. Agichtein, and A. Dubatovka, ‘‘What do you mean exactly? Analyzing clarification questions in CQA,’’ in Proc. Conf. Conf. Human Inf. Interact. Retr. Oslo, Norway: ACM, Mar. 2017, pp. 345–348, doi: 10.1145/3020165.3022149.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘‘BERT: Pre-training of deep bidirectional transformers for language understanding,’’ 2018, arXiv:1810.04805.

[16] K.J. Kopp, A.M.Johnson, S. A. Crossley, and D. S. McNamara, ‘‘Assess ing question quality using NLP,’’ in Artificial Intelligence in Education (Lecture Notes in Computer Science), vol. 10331, E. André, R. Baker, X. Hu, M.T.Rodrigo, and B. Du Boulay, Eds., Cham, Switzerland: Springer, 2017, pp. 523–527, doi: 10.1007/978-3-319-61425-0_55.

[17] Q. H. Tran, V. Tran, T. Vu, M. Nguyen, and S. B. Pham, ‘‘JAIST: Combining multiple features for answer selection in community question answering,’’ in Proc. 9th Int. Workshop Semantic Eval. (SemEval), 2015, pp. 215–219. Accessed: Feb. 22, 2024.[Online]. Available: https://aclanthology.org/S15-2038.pdf

[18] J. Lever, ‘‘Classification evaluation: It is important to understand both what a classification metric expresses and what it hides,’’ Nature Methods, vol. 13, no. 8, pp. 603–605, 2016.

[19] A. Humeau-Heurtier, ‘‘Texture feature extraction methods: A survey,’’ IEEE Access, vol. 7, pp. 8975–9000, 2019.

[20] S. Sarica and J. Luo, ‘‘Stopwords in technical language processing,’’ PLoS ONE, vol. 16, no. 8, Aug. 2021, Art.no. e0254937.

[21] A. I. Kadhim, ‘‘Survey on supervised machine learning techniques for automatic text classification,’’ Artif. Intell. Rev., vol. 52, no. 1, pp. 273–292, Jun. 2019, doi: 10.1007/s10462-018-09677-1.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ‘‘Efficient estimation of word representations in vector space,’’ 2013, arXiv:1301.3781.

[23] Q. V. Le and T. Mikolov, ‘‘Distributed representations of sentences and documents,’’ in Proc. 31st Int. Conf. Mach. Learn., vol. 32, Jan. 2014, pp. 1188–1196. Accessed: Feb. 23, 2024.[Online]. Available: http://proceedings.mlr.press/v32/le14.html?ref=https://githubhelp.com

[24] A. Wan, L. Dunlap, D. Ho, J. Yin, S. Lee, H. Jin, S. Petryk, S. A. Bargal, and J. E. Gonzalez, ‘‘NBDT: Neural-backed decision trees,’’ 2020, arXiv:2004.00221.

[25] L. Breiman, ‘‘Random forests,’’ Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/a:1010933404324.

[26] C. Park and F. Huffer, ‘‘How many trees in a random forest?’’ J. Korean Data Inf. Sci. Soc., vol. 33, no. 2, pp. 325–335, Mar. 2022, doi: 10.7465/jkdi.2022.33.2.325.

[27] M. P. LaValley, ‘‘Logistic regression,’’ Circulation, vol. 117, no. 18, pp. 2395–2399, May 2008, doi: 10.1161/circulationaha.106.682658.

[28] C.Cortes and V.Vapnik, ‘‘Support vector machine,’’ Mach.Learn., vol.20, no. 3, pp. 273–297, 1995.

[29] R. Kohavi, ‘‘Scaling up the accuracy of Naive–Bayes classifiers: A decision-tree hybrid,’’ in Proc. KDD, 1996, pp. 202–207. Accessed: Feb. 23, 2024.[Online]. Available: https://staff.icar.cnr.it/manco/Teaching/2005/datamining/articoli/nbtree.pdf

[30] Z. Huang, W. Xu, and K. Yu, ‘‘Bidirectional LSTM-CRF models for sequence tagging,’’ 2015, arXiv:1508.01991.

AUTHORS Profile

Mr. K. Jaya Krishna is an Associate Professor in the Department of Master of Computer Applications at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He earned his Master of Computer Applications (MCA) from Anna University, Chennai, and his M.Tech in Computer Science and Engineering (CSE) from Jawaharlal Nehru Technological University, Kakinada (JNTUK). With a strong research background, he has authored and co-authored over 90 research papers published in reputed peer-reviewed Scopus-indexed journals. He has also actively presented his work at various national and international conferences, with several of his publications appearing in IEEE-indexed proceedings. His research interests include Machine Learning, Artificial Intelligence, Cloud Computing, and Programming Languages. He is committed to advancing research and fostering innovation while mentoring students to excel in both academic and professional pursuits.

Mrs.Sk.Salma has received his MCA (Masters of Computer Applications) from QIS college of Engineering and Technology Vengamukkapalem (V), Ongole, Prakasam dist., Andhra Pradesh-523272 affiliated to JNTUK in 2023-2025