International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I3P214

Volume 70 Issue 3, 126-131, March 2022

Original Article

CQAs: Community Question Answering System Using Ensemble Learning Techniques for Job Assistantship

Venkateswara Rao P[1], A.P Siva kumar[2]

[1]*Research Scholar of Computer Science and Engineering, Jawaharlal Nehru Technological University, Anantapur Andhra Pradesh*

[2] *Associate Professor of Computer Science and Engineering, Jawaharlal Nehru Technological University, Anantapur, Andhra Pradesh*

pvenkat2004@gmail.com, sivakumar.ap@gmail.com

Abstract *- Individuals can search for and post queries or answers through community question and answer systems, which give a forum for exchanging information. It provides a set of responses with links to related queries for a freshly published query or query search system, which might be a long-term process to discover a meaningful answer. To address this, the system proposes a method for classifying the most appropriate and best answers from the archives using comparable queries. The model results reveal that Quora users and congested stacks are additional prospective to just discuss technology when compared to earlier open studies. It may fail if the query's keywords don't match the text content of huge papers containing pertinent inquiries about prevailing methods. Additionally, consumers are frequently not specialists and give confusing queries (Q&A) that produce mixed results and expose a flaw in current systems. To overcome these difficulties, researchers are rearranging the primary outcomes and proposing complex ensemble learning methodologies based on emerging technologies and platforms, as well as the amount of time and space required to develop the model.*

Keywords *— Stackoverflow, Quora, ML, Technologies, Social Media.*

I. INTRODUCTION

Software developers and programmers regularly utilise websites like Stack Overflow to get solutions to their questions. This information [1] can be used to figure out which programming elements and APIs are the most difficult to learn. In this post, we'll look at how to divide stack overflow issues into two categories: programming principles and the information requested. The flood assault contains a significant amount of data on a variety of computer programming issues. The person who submits the question must select specific tags that he believes are generally applicable for the context of the question to categorize it. The user has the option of selecting a tag from a list of existing tags created by additional users or creating a completely new tag. It is encouraged that users use existing tags whenever possible and only create new ones if they believe there is solid evidence that the issue raises a new topic that no one else has raised on this site. This makes it easier to find answers to upcoming questions. Nevertheless, the poster in question frequently displays many markings, and around is no mechanism to prioritize these markers now. Furthermore, the tags offered on Stack Overflow may possibly be unsuitable or extremely broad to contextualize the question, resulting in inappropriate requests to answer subsequent queries. Labels are subjective and open to consumer interpretation because they are assigned by consumers.

II. BACKGROUND KNOWLEDGE

In the realm of computing, software developers must study and collaborate using a variety of social collaboration platforms. GitHub and Stack Overflow are the most popular. Only search queries to retrieve relevant information from the GitHub repository or Stack Overflow Q&A [2] are supported by existing platforms. Abstract websites with technical questions and answers (Q&A) have emerged as promising sources of information on a variety of disciplines, with Stack Overflow being the most popular. Defending your rights Stack Overflow activates the game system to maintain user interest. The site motivates users by recognizing their engagement and contributions to the community and improving their contribution [3].

Question-and-answer websites such as Stack-Overflow are popular among programmers and academics, and they provide a wealth of information on the software, computing, and data development industries. Many people contribute to Stack Overflow, and many people read it. Stack Overflow employs a choice approach to ensure that good quality topic matter is easily visible. Users that provide high-level-quality answers or engaging questions are rewarded with positive returns and subject tags (for example, "C++", "Python"). Earlier people have dubbed it "Machine Learning." As a result, a "reputation" rating is created, identifying the most knowledgeable users in various categories. High-ranking

users are also given privileged privileges, such as voice voting, editing, and community moderating, based on their reputation score. Only well-documented, thematic, and appropriately marked questions/answers are allowed by the community moderators. These features not only make certain that the substance is of good superiority, but they also provide a valuable imagination for the platform's social analytics.

This section shows the number of questions and answers posted to Stack Overflow for each month from August 2008 through December 2020, as well as the number of active users who voted "in favour" of a historic stance. As a result of this, Stack Overflow has grown in popularity as a consistent resource everywhere users can acquire rapid answers to their questions that are based on computer programming and are more precise [3,4]. From a variety of views, Stack Overflow data is a study group. Such studies [1, 5, 6, 7, 8, 9, 10, 11] aided in determining the intensity of explicit scheduling of question-and-answer exchanges, as well as the extent to which these forums are filled as learning stages. When official documentation is lacking, relevant Stack Overflow comments are often used as a substitute for actual article documentation [12].

Stack Overflow, according to Parnin et al. [5,], has developed into a massive store of user-produced content that could supplement conventional specialist documentation. The documentation of interference, on the other hand, is haphazard, with little or no relationship or explicit connection to the API factors [5,13]. We haven't done much yet, but we can evaluate group documentation by various contributors and current pledges to comprehend the number of donations or to assess the value and nature of these bequests. Furthermore, traditional studies do not consider the intricacy of the scenarios while universalizing documentation for developers with varying degrees of understanding.

Even though a little research [14,15] has been done to advocate tags intended for Stack Overflow conversations, there is currently no viable method designed for endorsing learning materials to Stack Overflow discussion users. Given this, we're working on a method for recommending acceptable learning materials for Python and PHP, two popular programming languages. The main goal of this study is to create a proposal system that employs a controlled machine learning technique to supply developers with more wealthy discussion posts when they need them.

Stack Overflow has grown in popularity as a software development resource over the previous decade. Inexperienced programmers now turn to Stack Overflow for answers to questions about their software development projects [17-19]. Some programming language-related Stack Overflow queries may be missing a program design language label [20-21]. This could cause confusion among developers who are unfamiliar with all the common libraries available in a programming language. The problem of missing language tags could be remedied if entries are automatically labelled with their associated programming languages[22-23].

The goal of this investigation is to create a technique that can necessarily detect and recommend richer conversation articles to users of Stack Overflow repositories. By authenticating the information isolated from human perception with the machine, we were able to attain good results. In addition to the original product documentation, our framework offers users suitable guidelines. This piece is a follow-up to our prior article [16]. This work's key contribution can be summarized as follows: To create PHP and Python-related instructional materials, we use the knowledge resource available in Stack Overflow conversations [24-28]. Among software developers, Stack Overflow is the most popular Q&A website. Natural Language Processing (NLP) techniques can be effective in determining the programming language of source code files, and Stack Overflow's enquiries frequently contain a code excerpt as a platform for knowledge exchange and acquisition [28-32].

III. SAMPLE DATASETS

Our research examines developments in software development and technology use more than time by analysing user interactions and tags on Stack Overflow. It is thought that each user's interests and experience are determined by the tags linked to their posts and the reputation evaluation they obtain from posts that use these tags. The pairs can be used to create a user network with a community structure that reflects the user groups and technology. Trends can be investigated by looking at these networks across time. Thissegment explains exactly how the data was gathered and not much of the pre-processing processes that were carried out, such as associating content tags with "respond" posts and mapping the content tags. Each user's rating is based on the labels they've used over time. The essential components of the procedures for generating and evaluating the network are then described, including how each community is classified using its dominating labelsThe dominant tags connected with each community's members can be used to characterize it. The most common identifiers intended for the 16 populations discovered throughout the whole displays comprise the time-accumulated graph built in conjunction with a cumulative data point from 2008 to 2020 and the twelve-monthly graphs with temporal resolution. By focusing on the users, we can see how user populations work with technologies over time, how they obtain a score, and how they move around between technologies.

How do people earn a score and switch between different technologies? You can get a sense of the knowledge gatherings that variety up the larger software design and software advance environment by using the tags that come with each communal, such as which pieces of knowledge are stereotypically clustered together in practice, in what manner

new technologies emerge and are adopted, and how old types of machinery fade out of use, by using the tags that come with each communal.

A. System Architecture

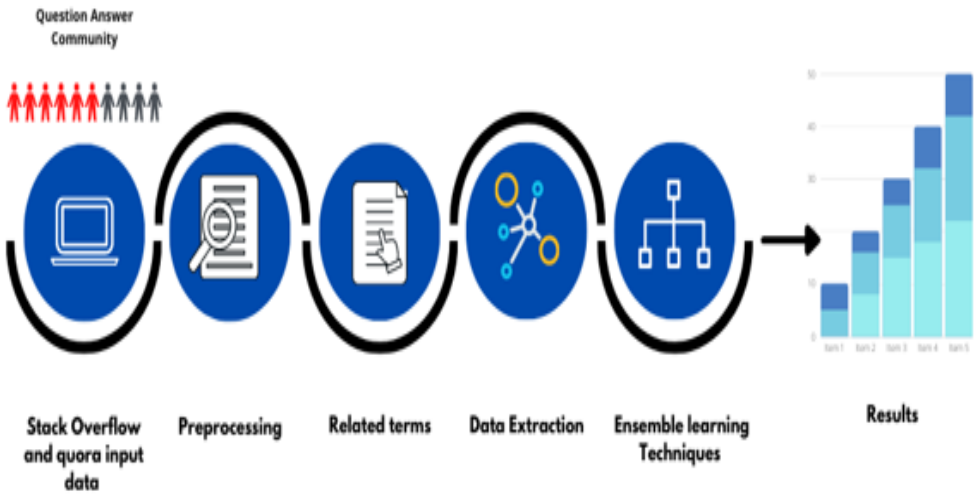

Fig. 1 system Architecture

From 2008 through 2020, a user-tag interaction graph was developed using all accessible data.

IV. METHODS

Ensemble techniques are ways for creating a single optimal prediction model by combining multiple learning algorithms or models. The model that was created outperformed the base learners on their own. Other uses of ensemble learning include feature selection, data fusion, and so forth. Bagging, Boosting, and Stacking are the three main types of ensemble approaches.

A. Bayesian Machine Learning

The goal behind the Bayesian approach is to use the Bayes' rule to incorporate certain previous opinions about the model into machine learning algorithms. When data is sparse or difficult to collect, as is often the situation in practice, it is quite valuable. Data D is not presumed to be correct in Bayesian analysis, but it is allowed to become "less wrong with the size." As more evidence is gathered, the process consists of iteratively updating our original belief or knowledge (prior) (data). The goal can be to discover the most likely model* (Bayesian inference) or to compute optimal predictions y* directly (Bayesian prediction).

In a previous tutorial, we discussed ensemble approaches. We'll look at two more ensemble learning methods in this tutorial: stacking and a mixture of experts. Both strategies are instances of meta-learning, which is while machine learning models are proficient using data derived from projections made by other machine learning reproductions.

a) Bayes' Rule

Let's say we have an empirical dataset D = {(x1, y1), ..., (xn, yn)} and a model θ. Then, by Bayes' theorem, we have

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)}$$

P(θ|D): posterior. What we look for.
P(D): normalizing constant independent of θ. So, it disappears when applying the Bayes' rule: P(θ|D) ∝ P(D|θ) x P(θ).
P(D|θ): likelihood. Model-specific, it is established by assigning higher likelihood to data with better model results.
P(θ): prior. Initial belief on how model parameters might be, specified in terms of parametrized distributions (uniform, normal, etc.). Inference should converge to probable θ.

B. Stacking

Using a variety of learning approaches, the models in the ensemble are stacked. The number of models developed is linked in order to calculate the definite expectation of any instance x:

$$\hat{y}(x) = \sum_{j=1}^{m} \beta_j h_j(x)$$

Stacking establishes a level-1 process known as meta-learner for learning the weights j of the level-0 predictors, though boosting chronologically computes weights j using an experiential formulation. That is, for the level-1 learner, the m calculations of each training case xi are now exercising data.

Although any machine learning technique can be utilised, least-squares regression is commonly used to solve the optimization problem.

$$\beta^* = argmin_\beta \sum_{i=1}^{n} \left(y(x_i) - \sum \beta_j h_j^{(-i)}(x_i) \right)^2$$

The leave-one-out prediction derived by working outhjon top of the subset of n-1 instances with the ith sample (xi, yi) left out is denoted by hj(-i). Base models hj are re-trained throughout the entire dataset and utilised to calculate formerly unseen instances x when appropriate weights j are estimated.

C. Boosting

It's an ensemble strategy in which each predictor learns from previous predictor errors to improve future predictions. The goal of boosting is to create a powerful learner from a group of vulnerable learners. Boosting works by changing the instruction set repeatedly established on the implementation of the previous predictions. It's a sequential ensemble method, in a sense. It gives higher priority to data points that remained in the past misclassified, causing succeeding standards to devote more effort to those more difficult data points. A prominent boosting method is AdaBoost (adaptive boosting).

a) AdaBoost

A binary classification dataset of n training samples (xi, yi), where yi is either +1 or -1, is shown in this section.

$$\hat{y}(x) = sign(\sum_{j=1}^{m} \alpha_j h_j(x))$$

The classifier weights αj are processed along with the proposition of offering a superior effect to the more exact classifiers. At to each one prototype prepetition, the misclassification error εj on the training set is intended:

$$\epsilon_j = \frac{\sum_{i=1}^{n} w_j(i)\, I(y_i \neq h_j(x_i))}{\sum_{i=1}^{n} w_j(i)} = \frac{\sum_{y_i \neq h_j(x_i)} w_j(i)}{\sum_{i=1}^{n} w_j(i)}$$

and used to compute αj:

$$\alpha_j = \frac{1}{2}\, ln \left(\frac{1 - \epsilon_j}{\epsilon_j} \right) > 0$$

Wj = (wj(1)..., wj(n)) probability weights are also employed to attribute varying significance to the training examples. After each model iteration, they are separately updated in advance, resampling the training set for the following classifier hj+1:

$$w_{j+1}(i) = w_j(i)\, e^{-\alpha_j y_i h_j(x_i)}$$

were, for binary classification,

$$y_i h_j(x_i) = \begin{cases} +1 & \text{if } x_i \text{ well-classified} \\ -1 & \text{if } x_i \text{ mis-classified} \end{cases}$$

The Weights are enhanced for cases misclassified by hj and dropped intended for samples that are well-classified since j > 0. The hj+1 training set has a higher percentage of misclassified points. In addition, the weights are normalised to produce Wj+1 a distribution with a sum of 1

$$w_{j+1}(i) \leftarrow \frac{w_{j+1}(i)}{\sum_k w_{j+1}(k)}$$

V. EXPERIMENTAL STUDY AND RESULTS

This section explains our research setting estimates the execution of our approach, and asks for a qualitative evaluation. A machine with an Intel Core i5 2.5 GHz processor and 4 GB of RAM was used to build the system. Python and MySQL were utilised for data processing, and WEKA was used for analysis. The front-end was created using Visual C#, and the back-end relies on a MongoDB NoSQL database. Using data from the Stack Overflow platform, we investigated the user populations formed around various technologies in the software development and computing industries. By associating users with the categories by which they collected the most "reputation," we generated user profiles that reflect their technology use and skill. We have been able to characterize each group established on its main skills/tools by looking at the prominent tags within each community.

By repeating the network construction and community detection process over a lengthy period of time, from 2008 to 2020, we were able to track the evolution of the Stack Overflow community. The tendencies revealed in this analysis are so similar to those in the digital industries that Stack Overflow can be considered emblematic of the greater software and computation industry.

Table 7. Comparison Analysis

Algorithms	CCI	ICCI	KS	MAE	RMSE	RAE	RRSE	TNI
Navie Bayes	0	19	-0.055	0.1.53	0.3136	102.853%	136.7825	19
AdaboostM	0	19	-0.055	0.1053	0.2425	102.853%	105.7657	19
Stacking	0	19	-0.055	0.1053	0.2367	102.853	103.2697	19

CCI	*Correctly_Classified_Instances*
ICCI	*Incorrectly_Classified_Instances*
KS	*Kappa_statistic*
MAE	*Mean_absolute_error*
RMSE	*Root_mean_squared_error*
RAE	*Relative_absolute_error*
RRSE	*Root_relative_squared_error*
TNI	*Total_Number_of_Instances*

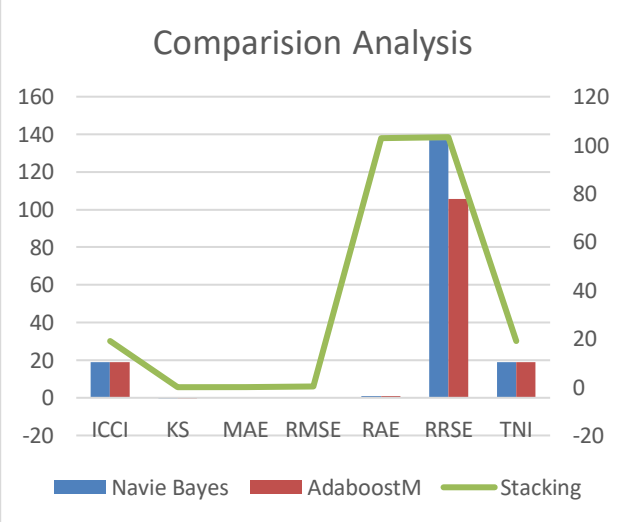

Fig. 8 Comparison statistics of ensemble Methods

VI. CONCLUSION

Question and answer (Q&A) websites offer a forum for communities to communicate and help one another. Stack Overflow is a standard programming-related Q&A website, with loads of developers seeking and providing helpful information. The Stack Overflow user community moderate's movement on the site and uses a support mechanism to promote high-quality content. We examine trends in the categorization of knowledge's and its users into distinct sub-communities using this data. Between 2008 and 2020, we analyzed all questions, replies, votes, and tags on Stack Overflow. Using a set of user technology interaction graphs, we used ensemble learning Technique algorithms to discover the largest user groups for each year, evaluate the technologies they use, how they are related, and how they evolve over time. Web development was one of the most popular and long-lasting groups. There is minimal mobility across communities because users either stay in the same group or do not gain any points. The popularity of various programming languages and frameworks has risen and fallen over time on Stack Overflow. These statistics provide information about the Stack Overflow user community as well as long-term trends in the software development industry. This project summarizes the global social structure of queries and responses, as well as the evolution of the key subjects of accommodation and occupation opportunities for next-generation knowledge's in real-time travel throughout the world for next-generation knowledge's.

REFERENCES

[1] Hu Y, Wang S, Ren Y, Choo KK. User influence analysis for Github developer social networks. Expert Systems with Applications. 15(108)(2018)108–18.

[2] Iraklis Moutidis, Hywel T. P. Williams, Community evolution on Stack Overflow,(2021) https://doi.org/10.1371/journal.pone.0253010.

[3] Ragkhitwetsagul C., Krinke J., Paixao M., Bianco G., and Oliveto R, Toxic Code Snippets on Stack Overflow. IEEE Transactions on Software Engineering, 1–1 (2019).

[4] Rossetti, Giulio, and Rémy Cazabet. Community discovery in dynamic networks: a survey. ACM Computing Surveys (CSUR) 51(2) (2018)1-37.

[5] Parnin, C.; Treude, C.; Grammel, L.; Storey, M.A. Crowd documentation: Exploring the coverage and the dynamics of API discussions on Stack Overflow. Ga. Inst. Technol. Tech. Rep (2012) 11. Available online: http://chrisparnin.me/pdf/crowddoc.pdf (accessed on 19 July 2020).

[6] Joorabchi, A.; English, M.; Mahdi, A. Text mining Stack Overflow: Towards an Insight into Challenges and Subject-Related Difficulties Faced by Computer Science Learners. J. Enterp. Inf. Manag. 29(2016) 255–275.

[7] Ponzanelli, L.; Bacchelli, A.; Lanza, M. Leveraging Crowd Knowledge for Software Comprehension and Development. In Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering, Genova, Italy,(2013)57–66.

[8] De Souza, L.B.L.; Campos, E.C.; Maia, M.d.A. Ranking Crowd Knowledge to Assist Software Development. In Proceedings of the 22nd International Conference on Program Comprehension, ICPC 2014, Hyderabad, India, 2–3 June 2014; Association for Computing Machinery: New York, NY, USA, (2014)72–82.

[9] Campos, E.; Souza, L.; Maia, M. Searching crowd knowledge to recommend solutions for API usage tasks. J. Software Evol. Process. (2016).

[10] A.Mallikarjuna, B. Karuna Sree, Security towards Flooding Attacks in Inter-Domain Routing Object using Ad hoc Network International Journal of Engineering and Advanced Technology (IJEAT), 8 (3)2019.

[11] Subramanian, S.; Inozemtseva, L.; Holmes, R. Live API Documentation. In Proceedings of the 36th International Conference on Software Engineering, ICSE 2014, Hyderabad, India, (2014)Association for Computing Machinery: New York, NY, USA, (2014) 643–652.

[12] Kim, J.; Lee, S.; Hwang, S.W.; Kim, S. Enriching Documents with Examples: A Corpus Mining Approach. ACM Trans. Inf. Syst. (2013) 31.

[13] Treude, C.; Barzilay, O.; Storey, M.A. How Do Programmers Ask and Answer Questions on the Web? (NIER Track). In Proceedings of the 33rd International Conference on Software Engineering, ICSE '11, Honolulu, HI, USA, 21–28 May 2011; Association for Computing Machinery: New York, NY, USA, (2011) 804–807.

[14] Souza, L.B.L.; Campos, E.C.; Maia, M. On the Extraction of Cookbooks for APIs from the Crowd Knowledge. In Proceedings of the 2014 Brazilian Symposium on Software Engineering, Maceió, Brazil, (2014) 21–30.

[15] Wang, H.; Wang, B.L.C.X.L.H.J.; Yang, M. SOTagRec: A Combined Tag Recommendation Approach for Stack Overflow. In Proceedings of the ICMAI 2019, Chengdu, China,(2019)146–152.

[16] Wang, S.; Lo, D.V.B.; Serebrenik, A. EnTagRec: An Enhanced Tag Recommendation System for Software Information Sites. In Proceedings of the IEEE International Conference on Software Maintenance and Evolution, Victoria, BC, Canada, (2014) 291–300.

[17] Baquero, J.F., Camargo, J.E., Restrepo-Calle, F., Aponte, J.H., Gonzalez, F.A., Predicting the programming language: extracting knowledge from stack overflow posts. In: Proceedings of Colombian Conference on Computing (CCC), (2017)199–221.

[18] Kennedy, J., Dam, V., Zaytsev, V, Software language identification with natural language classifiers. In: Proceedings of IEEE International Conference on Software Analysis, Evolution, and Reengineering, (19) 2016 624–628.

[19] Mallikarjuna Reddy, A., Rupa Kinnera, G., Chandrasekhara Reddy, T., Vishnu Murthy, G., et al., Generating cancelable fingerprint template using triangular structures, Journal of Computational and Theoretical Nanoscience, 16(5-6) (2019)1951-1955(5) doi:https://doi.org/10.1166/jctn. 201 9 . 7830.

[20] K. Kim, D. Kim, T. F. Bissyandé, E. Choi, L. Li, J. Klein, Y. L. Traon, FaCoY: a code-to-code search engine, in: M. Chaudron, I. Crnkovic, M. Chechik, M. Harman (Eds.), Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, 2018, ACM, (2018)946–957. doi:10.1145/3180155.925 3180187.

[21] R. Sirres, T. F. Bissyandé, D. Kim, D. Lo, J. Klein, K. Kim, Y. L. Traon, Augmenting and structuring user queries to support efficient free-form code 955 search, Empirical Software Engineering 23(5)(2018)2622–2654. URL https://doi.org/10.1007/s10664-017-9544-y.

[22] M. Liu, X. Peng, Q. Jiang, A. Marcus, J. Yang, W. Zhao, Searching StackOverflow Questions with Multi-Faceted Categorization, in Proceedings of the Tenth Asia-Pacific Symposium on Internetware – Internetware '18, ACM Press, Beijing, China, (2018)1–10. doi:10.1145/3275219.1020 3275227.

[23] P. T. Nguyen, J. Di Rocco, D. Di Ruscio, M. Di Penta, CrossRec: 1040 Supporting Software Developers by Recommending Third-party Libraries, Journal of Systems and Software (2019) 110460. doi: https: //doi.org/10.1016/j.jss.2019.110460.

[24] S. Baltes, L. Dumani, C. Treude, S. Diehl, Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts, in Proceedings of the 15th 905 International Conference on Mining Software Repositories, ACM, New York, NY, USA, (2018) 319–330. URL : http://doi.acm.org/10.1145/3196398.3196430.

[25] Kamel Alrashedy∗ , Dhanush Dharmaretnam, Daniel M. German, Venkatesh Srinivasan, T. Aaron Gulliver, SCC++: Predicting the programming language of questions and snippets of Stack Overflow, /The Journal of Systems and Software 162 (2020) 110505.

[26] Swarajya Lakshmi V Papineni, Snigdha Yarlagadda, Harita Akkineni, A. Mallikarjuna Reddy, Big Data Analytics Applying the Fusion Approach of Multicriteria Decision Making with Deep Learning Algorithms, International Journal of Engineering Trends and Technology 69(1)(2021) 24-28.

[27] Swarajya Lakshmi v papineni, A.Mallikarjuna Reddy, Sudeepti yarlagadda, Snigdha Yarlagadda, Haritha Akkineni, An Extensive Analytical Approach on Human Resources using Random Forest Algorithm, International Journal of Engineering Trends and Technology 69(5) (2021) 119-127.

[28] Venkateswara Rao, P., Kumar, A.P.S. The societal communication of the Q&A community on topic modelling. J Supercomput 78 (2022)1117–1143. https://doi.org/10.1007/s11227-021-03852-y.

[29] Ayaluri MR, K. SR, Konda SR, Chidirala SR. Efficient steganalysis using convolutional autoencoder network to ensure original image quality. PeerJ Computer Science 7(356) (2021). https://doi.org/10.7717/peerj-cs.356.

[30] A Mallikarjuna Reddy, Vakulabharanam Venkata Krishna, Lingamgunta Sumalatha and Avuku Obulesh, Age Classification Using Motif and Statistical Feature Derived On Gradient Facial Images, Recent Advances in Computer Science and Communications 13 (965) (2020). https://doi.org/10.2174/2213275912666190417151247.

[31] Alrashedy, K., Dharmaretnam, D., German, D.M., Srinivasan, V., Gulliver, T., SCC: automatic classification of code snippets. In: Proceedings of the 18th International Working Conference on Source Code Analysis and Manipulation (SCAM), (2018)203–208.

[32] P. Venkateswara Rao, A.P. Siva Kumar, An Efficient Novel Strategy For Online Social Networks Of A Q& A Community Forums Using Topic Modelling Methods (2021)21-34. DOI:10.17605/OSF.IO/F6JVP.